Optimizing the Coffee Experience by Developing a User-Centered, Internet Connected, High Precision Coffee Machine and Integrated System Experience

by

Jeremy Kuempel

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN MECHANICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2011

Signature of Author: ______________________________
Department of Mechanical Engineering
February 2, 2011

Certified by: ______________________________
Barbara Hughey
Instructor of Mechanical Engineering
Thesis Supervisor

Accepted by: ______________________________
John H Lienhard
Collins Professor of Mechanical Engineering
Chairman, Undergraduate Thesis Committee

Optimizing the Coffee Experience by Developing a User-Centered, Internet Connected, High Precision Coffee Machine and Integrated System Experience

by

Jeremy Kuempel

Submitted to the Department of Mechanical Engineering
On February 2, 2011 in Partial Fulfillment of the
Requirements for the Degree of Bachelor of Science in
Mechanical Engineering

ABSTRACT

The current state of coffee production is reviewed; from the origins of the plant grown to modern coffee brew techniques. Initial experiments are reported in which coffee was brewed at different temperatures for different lengths of time. The resultant drink was found to undergo changes in the objective properties of acidity and total dissolved solids (TDS), as well as changes in the subjective measurement of flavor depending on brew parameters. This discovery indicated that the flavor of coffee could be improved through precise control of coffee brew parameters, namely brew temperature and duration. A business model and internet-connected system for coffee brewing is presented. An automatic coffee machine that is capable of precisely controlling brew parameters was designed, manufactured, and tested. The machine showed potential by brewing two cups with exceptional flavor, but faced challenges with reliability and ease of use. Future work is planned to increase the reliability and reduce the cost of the machine so that it can be sold commercially.

Thesis Supervisor: Barbara Hughey
Title: Instructor of Mechanical Engineering

ACKNOWLEDGEMENTS

I would like to acknowledge Dr. Barbara Hughey for her boundless enthusiasm and support for this project. I would also like to acknowledge the friendly and helpful staff of MIT's Central Machine Shop and Pappalardo Laboratory, and Diecraft Machine and Engineering in Cincinnati. The machining and fabrication skills of the people at those shops were essential in giving physical form to this idea. I would like to thank James, Chris, and Caroline at Café Grumpy in Brooklyn for taking time out of their busy workday to talk with me about this idea. Their support in those early stages was invigorating and provided me not only with essential feedback on what direction to pursue, but also imbued me with new confidence for this project. I would like to acknowledge my mother, father, and brother for supporting me emotionally and financially during the long hours spent on this project. Although this idea began as my own, I was greeted by support and smiling faces at every turn of the road and would like to extend my deepest thanks to everyone who believed in me and assisted me. You have taught me to never underestimate the incredible power of saying "yes".

BIOGRAPHICAL NOTE

The author is a native of Cincinnati, OH and a graduate of Mariemont High School. He matriculated into the Department of Mechanical Engineering at the Massachusetts Institute of Technology in September of 2006. Since entering MIT, he has performed a variety of internships around the world. He spent a semester away from MIT to be an intern with the Pedestrian Safety department at Bavarian Motor Works in Munich, Germany from March until August, 2008. The following summer he interned at Continental AG in Yokohama, Japan. In the summer of 2010, he was an intern at Tesla Motors in Palo Alto, California and helped to develop the innovative touchscreen interface in the 2012 Model S sedan. Inspired by the possibilities he saw with touchscreen technology, he began designing the Blossom coffee machine and pursued the project in this thesis paper. He has been a brother of the Kappa Sigma fraternity since 2006.

The author can be reached by email at jkuempel88@gmail.com or by telephone at +1513-703-8604. Please contact him with questions or comments.

Table of Contents

1.0 Introduction

Coffee is a commonly drunk beverage that provides its drinkers with an energy boost. The full process of harvesting, roasting, and brewing coffee beans to produce the eponymously named drink is complex, with many steps that affect the flavor of the resulting drink. The initial stages of the process have been optimized by coffee producers such that the average consumer can purchase professionally roasted beans that are consistent in their quality and flavor. However, coffee brewing techniques remain either inconsistent or incapable of harnessing the full potential of the beans. Coffee has been found to contain more than 800 flavor components (twice as many as red wine)[1], indicating that careful brewing could reveal new tastes. Recent innovations in coffee brewing, especially the pod-brewing systems discussed in Section 2, have been focused on convenience and cost reduction, instead of quality.

In order to address optimization of quality and flavor of the coffee drink, a device that can be optimized for coffee flavor extraction by controlling variables associated with the brewing process has been designed and tested. The variables that affect the flavor of a coffee drink are time, temperature, and turbulence during the brew. In this project, time and temperature have been controlled while turbulence has been minimized. This device allows the effect of these variables on the attributes of the coffee drink, including objective measures such as acidity and total dissolves solids (TDS) as well as more subjective taste characteristics, to be tested. The results of coffee brewed with the Blossom automatic system have been compared to the results of coffee brewed with a standard manual technique in order to determine the relative advantages or disadvantages of the automatic system.

Note: In this paper, some confusion might result about the difference between referring to coffee beans, and coffee as a drink as both are commonly referred to as “coffee”. To avoid this confusion, the filtered and ready to drink beverage will be referred to as the “coffee beverage” or “coffee drink”. Clarifying modifiers will be utilized wherever necessary.

2.0 Coffee Background

There are four main factors that affect the flavor of a coffee drink: the attributes of the bean, the grind, the water quality, and the brew.

Two main species of coffee plants are grown for brewing coffee drinks: *Coffea Arabica* and *Coffea Canephora.*[2] Both species resemble a tall bush or short tree and will grow to approximately 30ft in height, but are commonly pruned to a shorter height for easier harvesting. After rainfall, the coffee plant sprouts white blossoms (shown in Figure 2.0.1) that turn into cherry-like seeds, which are picked and processed to produce coffee beans for roasting.[2] *C. Canephora*, commonly known as Robusta, has high caffeine content but weak flavors, and is primarily grown for usage in instant coffee, as flavoring for food products, or as filler for low priced coffee blends.[3] It is robust and easy to grow, especially at low altitudes.[3] *C. Arabica* is the most commonly brewed of the two and is known for having complex and rich flavors with lower caffeine content.[3] It accounts for 75-80% of global coffee production, although it is more difficult to grow and fares best at altitudes between 4000 and 5000ft.[3] Like wine, there are many varietals (a botanical term used for distinguishing different types of the same species) of *C. Arabica* grown worldwide that are noted for their unique flavors.[3]

Figure 2.0.1: Coffee blossoms and berries[2]

The roasted coffee beans that can be purchased in a store have a flavor that depends on the plant variety and region, as well as the roasting method. Like wine, coffee plants produce beans with flavor reminiscent of the region in which they were grown. Soil quality, altitude, and amount of rain and sun will affect the flavor of the bean. The way in which the bean is roasted will also change its flavor.[4] During the roast, the bean transforms from a light green color and ultimately reaches a dark brown or black, depending on the roast. Darker roasted coffees tend to taste like each other[4], with medium to light roasts allowing more of the regional flavors to come through, but each varietal responds differently to roasting. The temperature and time a bean is roasted will change its flavor, and the process has been perfected to the point where beans can be consistently produced with the same properties. This is the purchase point for whole bean coffee. Coffee enthusiasts insist that beans must be used within a week of roasting for optimum freshness, but modern packaging techniques can extend the period of freshness significantly.[5] However, once the coffee beans are exposed to oxygen, they typically have only a few days before their flavor is noticeably reduced.[4]

The beans must be ground and mixed with water to produce the coffee drink. Grinding the beans increases their surface area and allows the dissolution of chemicals into water to produce the flavorful drink. A good grind must be consistent, with all of the particles being roughly the same size. It is important to grind beans immediately before brewing and not leave them exposed to the air because grinding increases their surface area, thereby increasing exposure to oxygen. Inconsistencies in the grind will affect the rate at which the coffee dissolves into the water, and influence the flavor of the drink. The water quality is also important; coffee is mostly water, so some of the taste of the water used for brewing carries through in the drink.

Finally, the coffee must be brewed.

2.1 Brew Techniques

Brewing is the final step of the coffee-making process and is in principle as simple as mixing ground coffee with water. There are many types of coffee brewing devices, but almost all fit into one of the following six categories: drip brewers, espresso machines, pod brewers, full immersion, vacuum brewers, and percolators. The first four will be discussed in detail below. The last two, vacuum brewers and percolators, will not be discussed as they are not relevant to this paper. For all brew techniques, there are four main variables that affect the flavor of the coffee drink: the brew temperature, the dose (proportion of coffee to water), the immersion time, and the agitation or mixing during the brew.

The vast majority of coffee is made with drip brew systems such as the one shown in Figure 2.1.1, where heated water is filtered through ground coffee beans and stored in a glass carafe. In the most common form of this method, shown diagrammatically on the right side of the figure, a single heating element is used to both heat the water and keep the glass carafe at temperature. A one-way valve is placed upstream of the heating element to ensure that the near-boiling water travels up a tube to a distributor over the coffee grounds. The heated water then drips through the coffee grounds and into the carafe.[6] Automatic drip brewers are not known for producing good coffee, however. The process of heating water in all but the most expensive machines is not precise and frequently burns the top layer of the coffee grounds, then drops in temperature as it drips through and under-heats the bottom layer. Coffee produces different flavors at different temperatures (see Section 3 for details), so this temperature gradient is prone to extracting poor flavors as well as good ones. In contrast, manual drip brewers such as the Chemex (shown in Figure 2.1.2) are renowned for producing excellent coffee. Experienced baristas are able to heat water to the proper temperature and pour it such that the grounds come to temperature evenly. However, the grounds are still heated from the top initially and cool through convection such that the temperature of the bed is not controlled, a drawback inherent to the style of brew. One advantage of both automatic and manual drip brew systems is that cleanup quite easy; the conical filter can be lifted from the device and thrown away. A simple rinse of the rest of the equipment is usually sufficient.

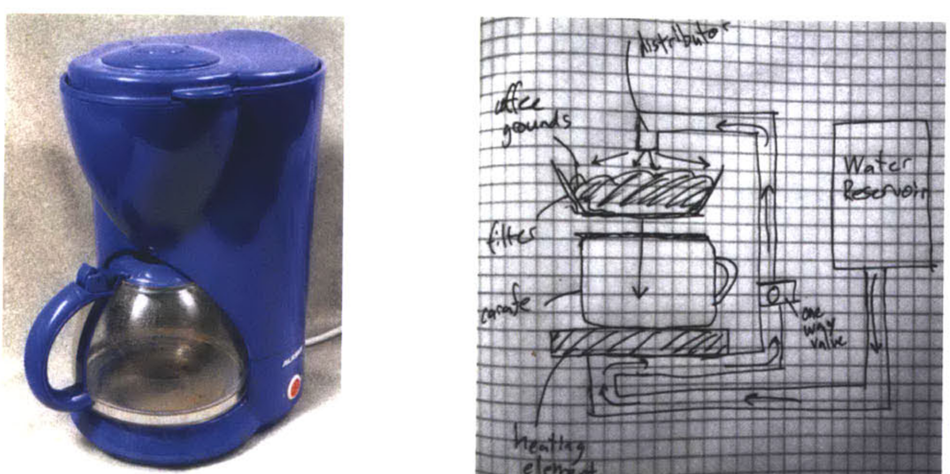


Figure 2.1.1: Photograph of commercial system[7] (left) and diagram of the brew system (right).

Figure 2.1.2: Chemex Brewer[8]

Figure 2.1.3: Rancilio Silvia[9]

Espresso is a style of coffee brewing invented in Italy where hot water is forced through compacted coffee grounds at pressure. The Rancilio Silvia, shown in Figure 2.1.3, is a popular home espresso machine. Coffee beans are ground to an extremely fine, almost dust-like consistency and are carefully tamped into an espresso basket. The basket is affixed to the espresso machine, which uses a pump and heating element to force hot water through the coffee. A typical shot of espresso takes between twenty and thirty seconds to pull at 9-10 atm pressure[3] and produces one fluid ounce of beverage. Espresso style coffee is renowned for its rich and complex flavor, and is more concentrated than drip coffee. However, the espresso brew process is notoriously difficult to control, with even slight variations in water temperature, pressure, coffee grind, and tamping resulting in large variations in taste.[3.2]

Pod brewers became commercially available in the USA in the late 1990s and have become increasingly popular in the last five years. Pod brewers utilize prepackaged coffee grounds sealed in proprietary containers with incorporated filters that are generally compatible only with the device made by the company. Although designs vary, most commonly require the user to insert the pod and press a single button to initiate the brew. The machine then heats water and runs a pump to force the hot water through the pod. After about a minute, the machine produces a single cup of coffee and the user can clean the machine simply by ejecting the pod. This technique of coffee brewing is very easy and fast, but is prone to several challenges. First, the prepackaged pods limit the choices of coffee bean to only those that are provided by the company, and are generally more expensive than buying whole beans. The pods also require coffee to be ground weeks before the coffee is brewed, which reduces freshness. Finally, the pods are generally made of plastic and produce more non-biodegradable waste than other brew techniques.

Total immersion coffee brewing is where coffee and water are mixed in a single volume. The French Press, shown in Figure 2.1.5, is one of the most common total immersion style brewers. The user operates it by mixing the coffee and water in a single vessel, traditionally made of glass, then using a metal filter to push the grounds to the bottom of the vessel after 3-5 minutes. The French Press is renowned for producing excellent coffee. However, it can be very messy and needs to be washed after every use. It can also produce inconsistent results, as both the initial temperature and brew time are determined by the consistency and skill of the user.

Another popular total immersion style brewer is called the Aeropress (shown in Figure 2.1.6). The Aeropress is a unique coffee maker that was invented by a Stanford lecturer who was seeking to design a simple brewing system that could make great tasting coffee. The Aeropress, like the French press, brews coffee through complete immersion, where the grounds and heated water are kept in the same volume for the entire brewing process.[11] Unlike the French press, it utilizes a plunger to pressurize the coffee and push it through a fine paper filter. Its plunger-in-body design allows for very easy cleanup; the plunger pushes out all the coffee grounds and the user needs to simply rinse the apparatus before making a second cup.

Figure 2.1.4: The Keurig B140[10] **Figure 2.1.5:** French Press[3.3] **Figure 2.1.6**: The Aerobie Aeropress[11]

2.2 Automated High Precision Coffee Brewing Systems

There are two full immersion brewers that can control brew temperature, duration, and agitation: the Clover coffee machine and the Bunn Trifecta. The Clover, currently owned by Starbucks and shown in Figure 2.2.1, is considered by some to be the world's best coffee machine.[12] The barista operates the machine by measuring out beans by weight, grinding them, and dropping them in the Clover brew chamber. Next, the brew parameters of dose, temperature, and duration are entered using the Clover's dials and LCD screen. The machine heats water to the specified temperature, adds the specified volume of water, and allows the mixture to steep for the specified period of time (generally about one minute). Usually the barista will stir the mixture gently during the brew. The Clover is very precise with its control of brew parameters and, although it is subject to variations in the stirring that the barista provides, is known for its reliability. One disadvantage noted by professional baristas is that the Clover, although fast, requires more coffee per unit of water than standard pour-over to produce comparably strong coffee.[13] Furthermore, the machine is no longer for sale to the general public as it is being produced exclusively for Starbucks stores.

The Bunn Trifecta (shown in Figure 2.2.2) is newer than the Clover; it was released in early 2010, while the Clover hit the market in 2006.[12] The Trifecta, like the Clover, is a full immersion brewer that can control brew temperature and duration. It is the world's only commercially available coffee machine, however, that can also control the agitation (stirring) of the coffee grounds during the brew. In an interview by the author with Caroline Bell, co-owner of the popular New York coffee shop Café Grumpy, the Trifecta was reported to produce a superior cup of coffee to the Clover.[14] She noted, however, that the machine gets little usage in her shop because of its time-consuming cleanup procedure and awkward appearance.

A final commercial grade automated high precision coffee brewing system is being developed by Luminaire Coffee, a company started by Olin College graduates. Luminaire produces a water distribution unit, shown in Figure 2.2.3, that can output a stream of water at a user-selectable flow rate between 4 and 12 oz/minute with a temperature within 0.2 degrees of the user's set point.[15] The machine is designed to be used in conjunction with a standard drip brew system and provides the barista with increased control over the brew process. However, this system is still susceptible to the issues associated with drip brewing.

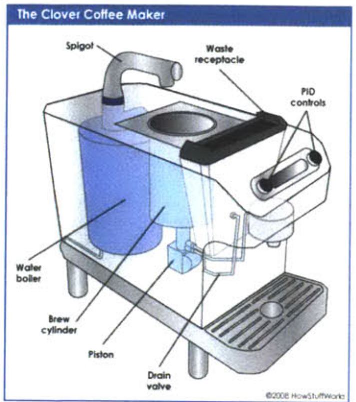


Figure 2.2.1: The Clover[16]

Figure 2.2.2: Bunn Trifecta[17]

Figure 2.2.3: The Luminaire Bravo[15]

3.0 The Effects of Temperature and Exposure Duration on Coffee Brewing

When investigating possibilities to improve coffee brewing, one must understand where opportunities for improvement are. Time, temperature, and turbulence while brewing are three factors that affect the flavor of coffee. In this section, measurements of the dependence of the properties of the coffee drink on the time and temperature are presented. Turbulence was not examined due to the added complexity of investigation. The work described in this section was performed by the author in fulfillment of the requirements of 2.671, "Measurement and Instrumentation" in the spring of 2010. It has been revised for this paper and is reproduced here because it became the inspiration for the topic of this paper, and provides a unique perspective on coffee brewing. The testing was recorded with metric units, and as such all discussion of temperature and volume will use metric. The machine, however, was designed using English units to allow ease of discussion with local machine shops and sourcing of parts from local distributors; discussion of dimensions will be done with English units.

3.1 Experimental Design

To investigate the effects of temperature and time on the process of coffee brewing, the Aeropress (see Figure 2.1.6) was utilized. It was chosen for this experiment because of its combination of consistent, full immersion brewing and easy cleanup. It is important in this experiment to control the type and grind of coffee being used, so only Folger's medium grind regular coffee was used. Each cup was brewed with 12.5 grams of coffee and 100mL of distilled water. A microwave was used to heat the water in a glass cup to avoid potential contamination from metal pots.

Acidity of coffee is frequently discussed when comparing brew styles, and the Aeropress website cites low acidity as a positive attribute of its coffee maker. The website states, "Because of the lower temperature and short brew time, the acid level of the brew is much lower than conventional brewers."[11] Acidity is therefore measured in this research to confirm or deny this statement and evaluate its relevance in discussion of coffee brewing and flavor.

Total Dissolved Solids (TDS) is a measurement of the dissolved salts and minerals in a fluid, and as such has the potential to be correlated to taste. The HM Digital COM-100 EC/TDS/TEMP measures the total dissolved solids in the coffee by using a 442 conversion from electrical conductivity. The 442 conversion was developed by and is a registered trademark of the Myron L Company and uses the conductivity of a fluid to calculate the parts per million of a mixture of 40% Sodium Bicarbonate, 40% Sodium Sulfate, and 20% Chloride.[18] HM Digital recommends using this conversion factor when evaluating coffee, and it is done automatically on the TDS meter. TDS is measured in this research to examine its relationship to flavor and if it changes based on the brewing parameters.

3.2 Utilization of Aeropress and Sensors

The Aeropress consists of three main parts: the plunger, the body, and the filter. The coffee is held in the body where it is mixed with hot water. The plunger applies a pressure to the liquid and pushes it through the filter into a glass cup in which the pH and TDS measurements were taken. The main components are shown in Figure 3.2.1.

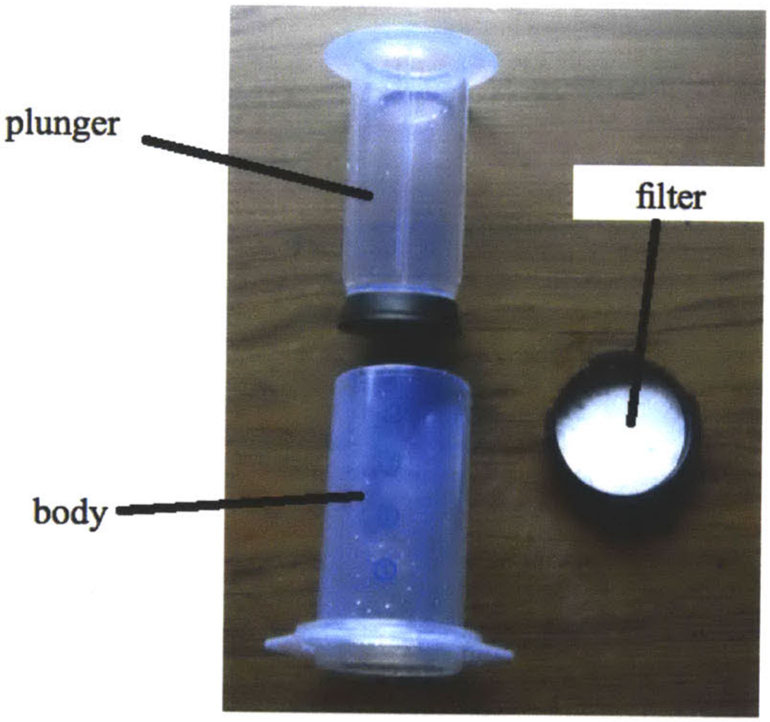


Figure 3.2.1: The three main parts of the Aeropress: the plunger, the filter, and the body/brew chamber.

A HM Digital pH-200 pH meter, a HM Digital COM-100 EC/TDS/TEMP combination sensor, and a Vernier Temperature probe were utilized. The acidity sensor and conductivity sensor are both fully equipped with readouts, and the temperature sensor is connected to a computer via a Vernier Labpro interface. Logger Pro was used to record the temperature. The sensors and experimental setup are displayed in Figure 3.2.1. The pH meter has a sensitivity of ±0.02, so all pH values presented in this paper are assumed to have this level of precision. The HM Digital COM-100 EC/TDS/TEMP measures the total dissolved solids in the coffee by using a 442 conversion from electrical conductivity.

The sensors were inserted into the coffee and stirred lightly. After allowing around 30 seconds for their measurements to settle, the numbers were recorded. A temperature probe was utilized to evaluate the water temperature before brewing. The sensors, Labpro, and computer are shown below in Figure 3.2.2.

Figure 3.2.1: Picture from left in front of the laptop computer: Vernier Labpro data acquisition unit, Vernier temperature sensor, HM Digital TDS sensor, HM Digital pH Sensor. The computer is a 2009 Apple Macbook.

3.3 Test Methods

It is important in this experiment to control the type and grind of coffee being used. For this reason, utilize preground coffee from Folger's was utilized to ensure that the type and grind of coffee was consistent. 12.5 grams of coffee and 100mL of distilled water per cup are used.

The goal of this experiment was to characterize changes in acidity and TDS as a function of brewing temperature and duration. To obtain this information, coffee was made with water at 80C, 90C, 95C, and 100C for durations of 1, 2, 3, 4, and 5 minutes. The coffee-water mixture was stirred for the first 10 seconds of the mixture and allowed to sit for the rest of the brew to control for possible effects of stirring on flavor extraction. The Aeropress was inverted to allow the coffee to brew in a single volume and flipped back over for filtering, as shown in Figure 3.3.1. After brewing, the coffee was allowed to cool below 80C and measurements were taken, as this is the upper temperature threshold at which the sensors could function.

The coffee was tasted by the author and evaluated in relation to previous cups as being more or less flavorful than other cups next to it in brew temperature and duration. The author sampled every cup in this test, and many more that were not measured, so he developed a strong understanding of the flavors and tastes that are possible to be extracted from Folger's pre-ground classic roast medium coffee. He found that there is a certain range of flavors that can be extracted from this particular brand of coffee, and was able to recognize the proper flavors as being present in the coffee and in the right proportion. When evaluating the flavor of the coffee he looked for mouth feel, balance of flavors, and good aftertaste.

Using his experience with this particular brand of coffee, the author was able to evaluate each cup as being more, less, or approximately equal to the other nearby cups in quality of flavor. For example, coffee brewed at 100C for 5 minutes was compared to coffee brewed at 100C for 4 minutes as well as coffee brewed at 95C for 5 minutes. In this way, the author was able to evaluate the coffee flavor with confidence.

It is important to note, however, that the comparisons regarding taste are subject to the preferences of the author. A more scientific and objective test would involve a wider selection of tasters so that more taste preference data could be gathered and analyzed. Unfortunately, there was not time to perform such testing during this project. Information regarding the flavor of coffee is of particular interest to this research, however, so the taste preferences collected by the author will be discussed.

Additional testing was done to find a more precise characterization of the change in pH as a function of temperature. The coffee was brewed for twenty seconds with stirring the whole time. After 20 seconds, the apparatus was inverted and the plunger was used to empty the mixture into a glass cup. The temperature of coffee was varied in eight intervals, starting at 100 °C down to 42.3 °C. After brewing, measurements of pH were taken.

Figure 3.3.1: Inverted Aeropress for brewing coffee (left) and ready to filter into cup (right).

3.4 Results from Testing

The pH and TDS of the coffee were measured for 26 different variations in temperature and brew duration. The results were plotted three dimensionally using MATLAB to investigate trends across both parameters. Also, pH was plotted against brew temperature at constant duration to show their relationship. Table 3.4.1 shows all recorded values for pH and TDS at a variety of temperatures and durations. TDS values are missing from some cells; in these locations TDS was not recorded due to experimental errors during those particular tests. Refer to this table to note the exact data that was recorded for each point, it will be discussed through the use of graphs in the rest of the paper.

Table 3.4.1: Data from manual testing performed with the Aeropress.

Temperature	Duration (s)	pH	TDS
100	20	5.04	
100	60	5.12	3440
100	120	5.14	3680
100	180	5.15	3590
100	240	5.11	3640
100	300	5.17	3040
95	60	5.18	3370
95	120	5.21	2900
95	180	5.19	3770
95	240	5.22	3510
95	300	5.27	3580
91	60	5.05	2380
91	120	5.08	3070
91	180	5.1	3900
91	240	5.08	2810
91	300	5.09	3070
85	60	5.19	3350
85	120	5.2	3170
85	180	5.18	2830
85	240	5.16	3190
85	300	5.14	2800
86.9	20	5.06	
82.5	20	5.08	
80	60	5.18	2960
81	120	5.22	3550
81.3	180	5.23	4150
81	240	5.19	3340
81.5	300	5.18	2960
72	20	5.09	
58.2	20	5.14	
53	20	5.16	

3.5 Acidity and TDS as a Function of Temperature and Duration of Brew

The data listed in Table 3.4.1 is graphed three dimensionally in Figures 3.5.1 and 3.5.2. To find a pH or TDS level from these figures, the reader must select the temperature of interest on the y-axis, then move along the x-axis until finding the brew duration of interest. Once the point of interest has been found, the color at that point corresponds to a pH or TDS level as shown in the color bar on the right of the figure. For example, at 95 degrees Celsius and 180 second brew duration, an orange-red color can be found on the TDS graph. After referring to the color bar on the right, this orange-red color can be noted to be between 3600 and 3800 ppm, or around 3700 ppm. Just to double-check with Table 1, this point does in fact correspond to a TDS level of 3770ppm.

The region of best flavor as determined by the author was circled on the graphs as well.

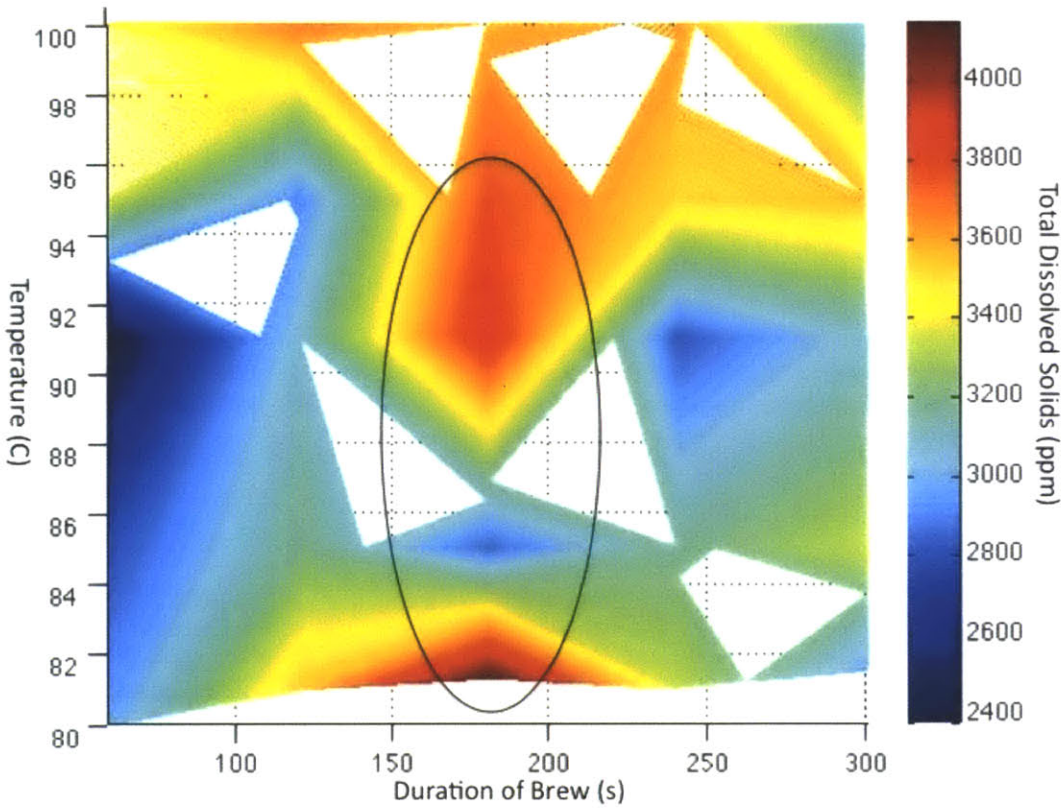


Figure 3.5.1: TDS as a function of temperature and duration. The circled region corresponds to the area with best flavor.

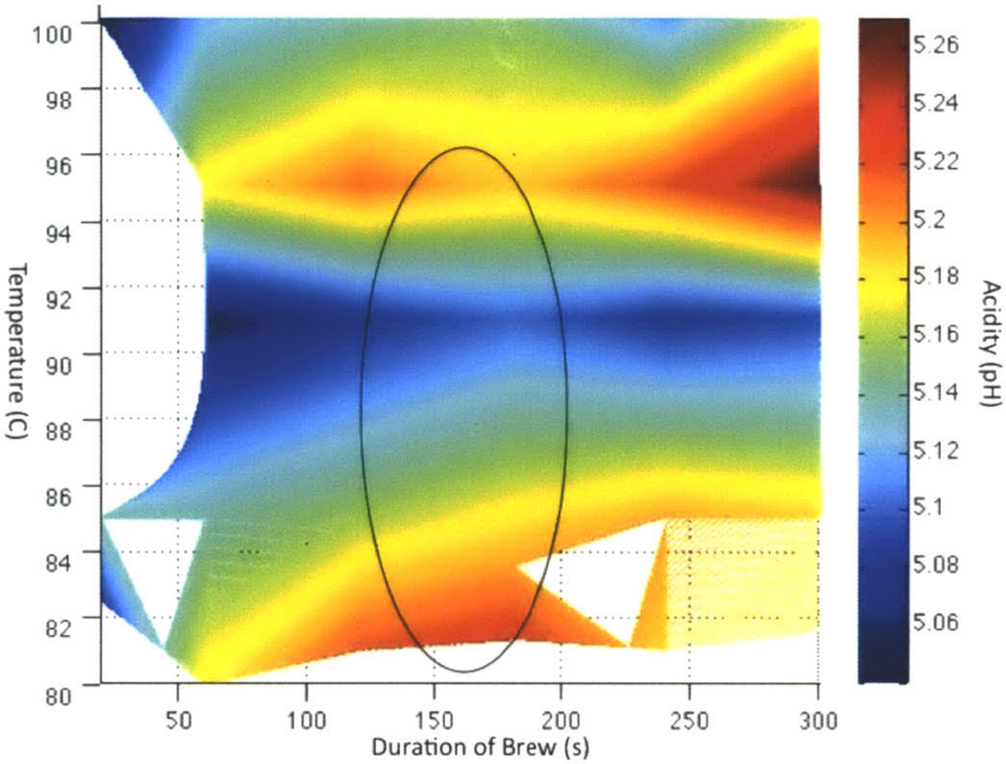


Figure 3.5.2: pH as a function of temperature and duration, the circled region corresponds to the area with best flavor. Note that there is no apparent correlation between good flavor and pH.

Plotting as shown in Figs 3.5.1 and 3.5.2 shows how coffee attributes change dynamically as a function of both parameters. As can be seen in Figure 3.5.2, pH shows primarily horizontal lines of color, demonstrating that pH is more strongly related to water temperature than duration of brew. TDS, on the other hand, has regions of similar TDS levels that are a function of both temperature and duration. However, when comparing Figure 3.5.1 to Table 3.4.1, one can realize that the "zones" of common TDS are partly a result of MATLAB coloring in around data points. There are several points such as the TDS values at 240s, 91C and 180s, 85C that seemed to be out of line with the other TDS values recorded in that area. More testing would be required to create a higher resolution graph, so that the cause of this situation could be investigated. It is possible that these points are an indication of a trend, but it is also possible that they are the result of inaccuracies in the testing and contamination of samples. With the exception of the TDS value at 180s, 85C, high values of TDS were found to correlate to good flavor.

Figure 3.5.3 shows a graph of pH data for a range of temperatures brewed for a duration of 20 seconds. This analysis of data from a constant 20s duration brew shows a consistent trend. A 4% difference was found between the pH of coffee made with near boiling water and that of coffee made with warm water. Coffee made with hot water is more acidic than that made with cooler water, but the overall change is not drastic in the range in which people usually make coffee.

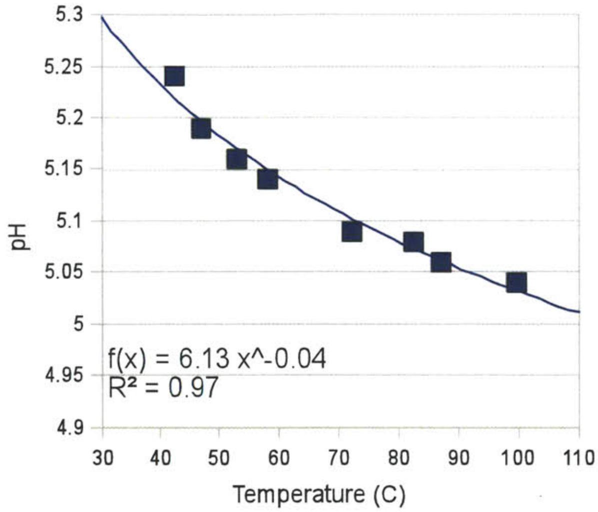


Figure 3.5.3: Curve fit of pH data for a range of temperatures at 20s brew time.

3.6 Discussion of Possible Errors

Despite the measures taken to control variables and error, there are some areas where the test equipment or materials could have had an effect on the results. Coffee aficionados cite fresh coffee as having better flavor than coffee that has been exposed to air for a period of time longer than a few days. To control this variable in future research, all data should be gathered with freshly opened coffee. However, in this experiment, bulk containers of Folger's coffee were utilized to save cost. The containers were closed between rounds of testing, but nevertheless the author noted deterioration in flavor as the coffee aged and was exposed to air. The effect of coffee age on pH, TDS, and flavor was not investigated in this experiment, although it may have impacted the results. Distilled water was used for all the tests, and measurements of the initial TDS level of the water always confirmed it to be below 10ppm. However, the pH sensor will not function such a low TDS level, so initial pH was not measured. The pH of distilled water is known to change from absorbing CO_2 from the air[19], so it is possible that the initial pH of the water used for the tests presented in Table 3.4.1, which were performed over several weeks, could have varied. The information presented in Figure 3.5.3 was gathered over the course of a few hours, so the acidity of the initial distilled water used should be consistent. Furthermore, although care was taken to mix the coffee grounds with water for a set period of time, there was a potential error of approximately 10 seconds when separating the grounds from the water by flipping over the Aeropress and depressing the plunger.

3.7 Summary of Results from Manual Testing

The effects of water temperature and duration of brew on coffee pH and TDS were investigated. These characteristics were expressed as a function of both water temperature and brew time. It was shown that pH is correlated to initial water temperature, but was not significantly impacted by brew duration. Good flavor was found to be correlated to TDS, with the exception of one data point.

This research served to underscore the importance of coffee brewing on coffee flavor. The Folger's coffee that was used in this experiment exhibited drastic changes in quality as the result of different brew styles. At its best, it was rich and well balanced, with chocolatey and earthy undertones. At its worst, it was bitter, watery, and metallic. This experiment has shown that coffee flavor and TDS change significantly as a result of the brewing technique, but not necessarily acidity. Raising the temperature will make the brew more acidic, and allowing it to brew longer will increase TDS—up to a certain point. Allowing it to brew too long will actually cause the TDS to decrease and diminish the flavor.

A major development was the model of considering coffee flavor as a function of both temperature and duration of brew. Coffee aficionados should consciously vary these parameters when making the beverage and investigate for themselves how a certain bean responds to changes. They should try to develop an intuition for how a change in duration or temperature will change the taste. This research has shown that increased control on coffee brewing parameters can and will open new possibilities for coffee taste. To design a truly excellent coffee brewing system, however, more than just precision of brew control would need to be taken into account. The brewer needs to be designed specifically for the intended user.

4.0 User Study for Design of Blossom Machine

A user study was conducted in Manhattan and Brooklyn in New York City. Five different, well-respected cafés were visited and studied to gain insight into how to build a better coffee machine. The cafés in Manhattan were Abraço (86 E. 7th St., New York, NY 10003) and Intelligentsia (594 Broadway, Suite 909A, New York, NY 10012) were visited. The cafés in Brooklyn were Oslo (328 Bedford Ave, Brooklyn, NY 11211) Blue Bottle Coffee (160 Berry St., Brooklyn, NY), and Café Grumpy (373 7th Ave, Brooklyn, NY 10011) were visited. Input was gathered both for a machine designed for usage in a café and for home use. Table 4.0.1 contains a list of the needs that were discovered and shows their place of discovery.

Table 4.0.1: User needs and their place of discovery

User Need	Abraço	Intelligentsia	Oslo	Blue Bottle Coffee	Café Grumpy
Hand-made feel	x		x	x	
Reliability					x
Price			x		x
Good Ergonomics		x	x		x
Good Aesthetics					x
Precision		x			
Efficiency					x
Ease of Self service					x
Speed of brew					x

Abraço, Blue Bottle Coffee, and Café Grumpy all had four people running the shop, while Oslo had only one person. Each barista had to be efficient and able to do multiple things at one time. In Blue Bottle Coffee, a barista was seen making five cups of pour-over coffee at the same time by having all the cups next to each other and pouring water into them sequentially. During an interview with Intelligentsia, it was noted that some espresso machines have sharp corners that baristas could run into and get hurt. A better coffee machine would have good ergonomics that are designed specifically for the café. The barista at Oslo noted that the physical size of their coffee machine was a consideration when they purchased it; they wanted a machine that would not take up too much space on their cluttered countertop. He also noted that they bought a manual espresso machine because they felt it allowed more control over the brew. The barista at Blue Bottle Coffee stated that he preferred the hand-made aspect of traditional pour-over coffee as opposed to the automatically controlled Clover. Although only Café Grumpy had a Clover machine, all the baristas interviewed were familiar with the device.

More time was spent interviewing people at Café Grumpy than the other cafés. At Café

Grumpy, it was noted that the physical shape of the Clover machine allows for some interaction between the barista and the customer during the brew, an attribute that they've found to be helpful. The baristas interviewed at Café Grumpy stressed the importance of speed and allowing the barista to do other things while the machine was working. Efficiency was important as well: they specified that, for them, the perfect machine would be able to brew 60g of coffee per 1000mL of water so that they could save some cost over the Clover, which requires 90g of coffee pre 1000mL of water. They wanted to be able to make a good-tasting cup in less than two minutes of total interaction time, and noted that the time required to serve a customer was a direct cost for them. They did not share the same sentiment as the baristas at Oslo and Blue Bottle Coffee towards manual systems, noting that those devices required a barista to watch them as they worked, which distracted the barista from performing other revenue-generating activities. Chris, the co-owner of Café Grumpy, gave four recommendations for a coffee machine.[14] First, he was concerned about the price of the machine. Second, it needed to work reliably and be high quality. Third, the machine needed to work quickly. Finally, the machine should have good ergonomics and aesthetics that fit well in the shop. A hot water self-rinse feature was also requested so that the machine could switch between brewing different beans without mixing the flavors.

Customers at Abraço were also interviewed regarding their coffee brewing preferences. The customers that were encountered at this high-end shop were coffee enthusiasts, and may not be representative of the population at large. However, they might also more likely to be early adopters of a new coffee brewing system. One customer noted that he brewed his coffee using a French press because he preferred the flavor, and enjoyed the process required to use the device. When asked if he would use a machine that required only a single button press to produce coffee of equal or better flavor, he said that he wouldn't use it. Another person noted that he used his French press because it made enough coffee to fill his 16oz mug.

A common sentiment among the customers interviewed was that it was difficult to find a cup of really good coffee in New York City. At Abraço, two customers noted that they traveled out of their way to this coffee shop because they felt the product was better. In a pizza shop, a man was overheard complaining about how difficult it was to find good coffee. This sentiment indicates that New York City could be a good place to introduce a new type of coffee machine.

4.1 A User-Centered, High Precision, Gourmet Coffee Machine: The Blossom Coffee Machine

A new coffee machine, called the Blossom, was designed while taking into account the results from the user study and benchmarking analysis. The new machine needs execute high precision control of brew variables, namely temperature, agitation, and duration. A fourth variable that was not discussed earlier is brew pressure. The effect of pressure on full immersion coffee brewing has not been publically studied partly because there is currently no commercially available machine that can brew coffee in that style and also control pressure. However, it was noted by James Park Brannon of Café Grumpy that barometric pressure could affect the flavor of coffee.[13] Therefore, a machine that could control the pressure of brew as well as time, temperature, and turbulence would have a distinct advantage over existing systems. By controlling this fourth variable, it could be possible to truly recreate the same cup of coffee time after time, regardless of barometric pressure or altitude differences; something that conventional machines still struggle to do. It would also create the possibility of experimentation at higher than atmospheric pressures.

Providing a superior brewing system would be interesting to coffee enthusiasts, but to bridge the gap between early adopters and coffee geeks, the machine needs to be able to provide more than just a great cup of coffee. It was discovered during the user study that cafés and private customers have different needs and preferences when it comes to brewing coffee. Therefore, the design of the machine needs to be intimately intertwined with its use case scenario. Cafés and private customers were interviewed during the user study, but the workplace is another significant market for coffee machines. Taking these three use case scenarios into account, a superior coffee machine is not a single machine for everyone, but rather three different machines suited for three different scenarios. Table 4.1.1 lists the specifications for the Blossom system and their relative importance in three different scenarios.

Table 4.1.1: User needs and their relative importance in three different use case scenarios. All of the needs are important for these scenarios, however 1 indicates the highest importance and 3 is the least important.

Blossom Specification	Café	Workplace	Home
Hand-made feel	2	3	1
Reliability	3	1	2
Price	2	2	1
Good Ergonomics	1	2	2
Good Aesthetics	1	3	2
Precision	1	3	2
Efficiency	1	2	3
Ease of self service	2	1	3
Speed of brew	1	1	3
Connectivity	1	1	1

4.2 Blossom as a System Experience

With control over four brew parameters, there are a staggering number of possibilities for brewing any particular bean. So many, that new customers would have to work through a steep learning curve before they could consistently make good coffee. The number of possibilities explored by any given person would be directly related to their interest in coffee brewing, and it would be time consuming to personally explore all the possibilities of even a single type of bean. For this reason, it is important to equip the Blossom coffee machines with internet connectivity and the ability to report on the results of coffee brewed with different brew parameters, possibly through the use of a touchscreen.

Incorporating internet connectivity in the Blossom coffee machine opens a variety of interesting possibilities. Customers could enjoy their favorite cup of coffee in any of the three places simply by using the same type of bean and the same brew parameters. They could connect to an online account through the machine itself and have their specific parameters downloaded to make their cup. When all the machines are connected to the internet, they can also all upload their coffee information to a single database, making it easy for customers to search for their beans and find brew recommendations. Internet connectivity allows machines to be updated with new brew parameters and firmware improvements, and also to report their usage information and operational status. Internet connectivity on the coffee machine also creates new potentials for revenue streams, such as direct sale of coffee and directed advertising. A successful integration of internet connectivity with coffee machines would represent a new stage in the development of internet usage, moving it from computers and smartphones into commonplace equipment. Further discussion of the possibilities presented by internet connectivity is included in Appendix A.

4.3 Blossom in the Café

In the café, the barista is busy, distracted, and needs the machine to do its job and get out of the way. The perfect coffee machine for the café would require a very short interaction to set up and get started, the shorter the better. In order to win over experienced, traditionalist baristas, it is important for them to still feel engaged with the coffee brewing process while providing timesaving and qualityenhancing benefits. The baristas at Café Grumpy and Oslo reported that they service their machines regularly, so the Blossom should be designed to allow customers to do so. However, it was noted at Café Grumpy that the Trifecta requires a time-consuming cleaning after every cup, so service requirements should be coordinated such that they could be performed without taking up time during the workday. The industrial design of the machine is important as well. It needs to provide for a good interaction between the barista and the customer, as well as looking at home in the coffee shop. However, the look also needs to be unique and recognizable to consumers so that they can spot the machine and recognize its brand. Efficiency is important to café owners as well, so the machine should not require a large amount of ground coffee to brew a good cup.

It was found during the user study that each café has unique needs and opinions on coffee brewing. Requirements such as "aesthetics" or "ergonomics" are specific to the café the machine is being used in. Although a coffee machine could be designed to fit a variety of cafés, the approach would require making compromises on specific attributes to fit a larger scope. Advantages can be found by designing a coffee machine and store experience in tandem, such that the attributes are intertwined. Discussion of synergies between store and machine design is included in Appendix A.

4.4 Blossom in the Workplace

For the workplace, speed of brewing is just as important as in the café, and reliability even more so. In the workplace, coffee machines are used all day and receive even cursory service at irregular intervals, if at all. The ideal machine for the workplace could make a single cup of coffee in less than a minute, and be able to consistently make cups without any service for as long as possible. Employees should be trusted to do certain aspects of service, and the design of the Blossom should take that into account and make such tasks simple and straightforward. Taste of coffee and control of variables is less important in the workplace, but the machine should still produce coffee that tastes better than competitive systems.

In the workplace, people have three main goals when they get coffee: to take a break from work, to socialize with coworkers, and, of course, to get some coffee. They are often stressed with work, short on time and patience. The ideal machine would have a superior user experience that is easy and enjoyable to use. It would be incredibly reliable so that they could plan a break of only a few minutes around it. It would also be honest by communicating with the user how long tasks will take and its current state of functioning. It would not automatically perform functions that could delay users from getting their coffee. It would also serve as a conversation piece, inspiring coworkers to interact with each other and promoting inter-office relations.

4.5 Blossom in the Home

The machine for home use has the lowest speed requirement of the three machines, but needs to be extremely reliable. Cost, industrial design, and user experience are crucial for home use. The coffee should also feel personal. It was a common sentiment during the user study that manual machines were preferred because the users felt they had more control over the process. Therefore, this machine would need to be an aid to the user and something they interact with, something that helps them to make their coffee rather than simply doing it for them. In order to create this sort of interaction, the machine should control the brew parameters as the user specified. However, the machine should not grind the beans or have them pre-inserted. There was a study for cake batter where users were asked which cake they preferred between two types of batter. Double blind taste tests had confirmed that the two batters had identical flavor; the only difference was that one of the batters required people to add eggs, while the other only required the addition of milk. People who baked the cakes themselves had an overwhelming preference for the batter that required them to add eggs, even though it required more work from them than the other batter. By requiring the user to select their own beans, grind them, and insert them into the machine, Blossom can capitalize on the learning from this study. The user would likely smell the beans prior to brewing them and prime their olfactory nerves with the impending taste of coffee. They would start thinking about coffee and reminisce about previous cups they had had, and be better prepared to enjoy a new cup. Double blind taste tests will not tell the whole story of a cup of coffee: a superior user experience will lead to superior user satisfaction, even with equivalent taste. Having both a superior user experience and a superior taste will give the home brewer a lead on competitors.

5.0 Design Process

A laboratory-grade test piece was designed, one that could be used to demonstrate the principles of temperature controlled full immersion brewing. It must be able to hold a constant temperature during the brew, within 1 degree Fahrenheit. It should also be able to control the duration of brew within 3 seconds. It does not need to be attractive or incorporate any other functionality, simply to automate the test process used in Section 3. With an automated process, additional research can be done into the flavor potentials of coffee and it can be used as a way to verify and prove the foundation of the concepts stated in Section 4.

5.1 System Brainstorming and Final Design

A design option that was examined was to simply take an Aeropress and mount servos such that the movement of the plunger would be automated, as shown in Figure 5.1.1. This design was eliminated over concerns that it would not brew a large enough volume of coffee, and would have been limited in its ability to incorporate future innovation.

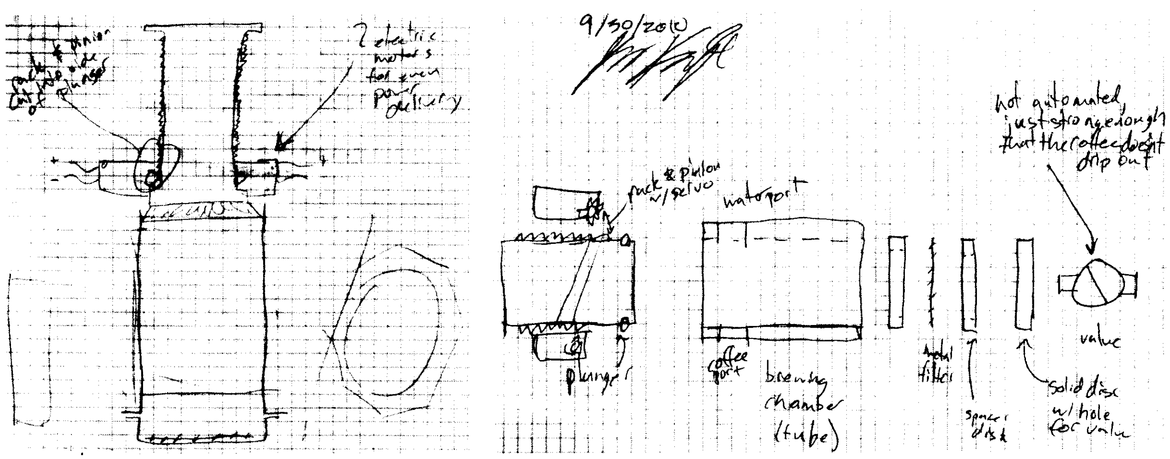


Figure 5.1.1: An early design (left) and exploded view (right)

The final design consisted of a single vertical cylinder made from stainless steel with a plunger at the top end and a filter at the bottom end as shown in Figures 5.1.2, 5.1.3, 5.1.4. This design allows for a heating element to be wrapped around the brew chamber for temperature control, and for the plunger to regulate the duration of brew. The overall structure is durable, with modular components that provide a flexible platform for future innovation.

Figure 5.1.2 shows the first version of the system, which has a vertically oriented cylindrical brew chamber with a filter at the bottom end and plunger at the top end. Many updates were made between this first model and the final version shown in Figure 5.1.3, which are discussed in each components section. The assembled device is shown in Figure 5.1.4, and Figure 5.1.5 shows the assembled machine with labels of its components.

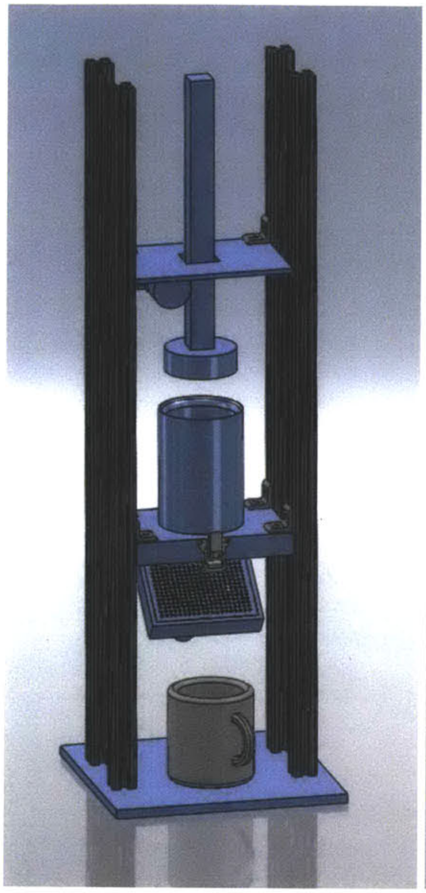
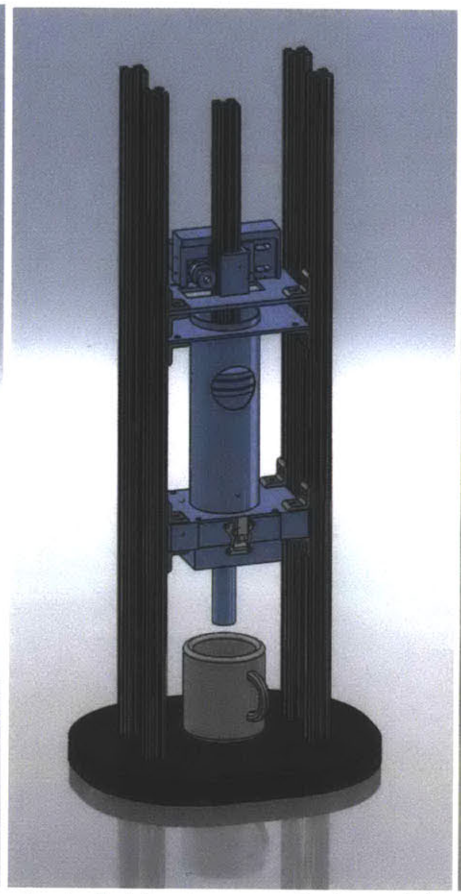
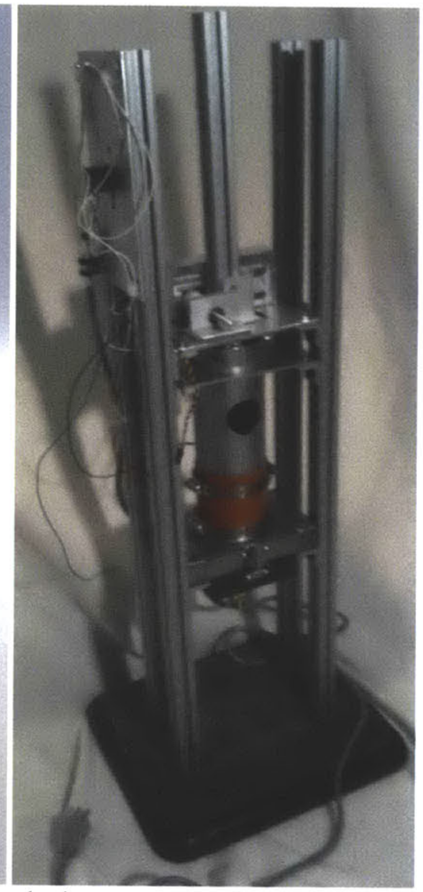




Figure 5.1.2: (left) Early design. Included to show evolution of the design.
Figure 5.1.3: (center) Final design. Note the change of having the plunger stay within the brew chamber at all times, and addition of a hole in the front of the brew chamber.
Figure 5.1.4: (right) Assembled Machine. The control electronics were not shown in the computer model.

The user will insert both coffee and heated water in through the hole in the front, and program the desired temperature and duration of brew using an Arduino microcontroller programmed via USB interface from a laptop computer. A thermocouple submerged in the brew provides feedback to the computer, which executes proportional-integral-derivative (PID) control of brew temperature by controlling power to the heating element wrapped around the brew chamber. After the user-specified brew duration has been reached, the servomotor actuates the piston using a rack and pinion system and pushes the coffee through a filter in the bottom of the system and into the user's cup.

The five main parts to the system are the support structure, the plunger assembly, the filter assembly, the brew chamber, and the control electronics. They are discussed in the following sections.

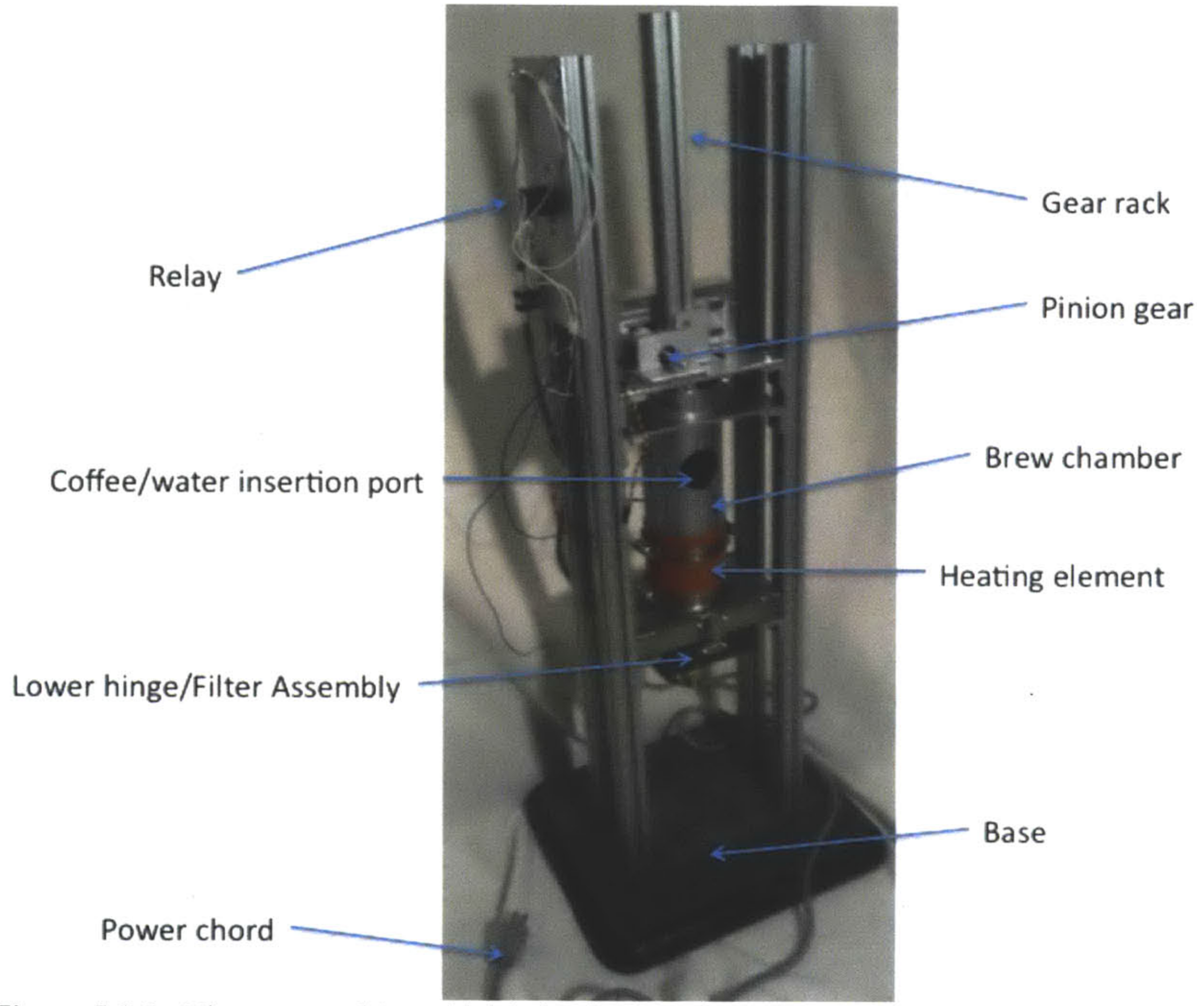


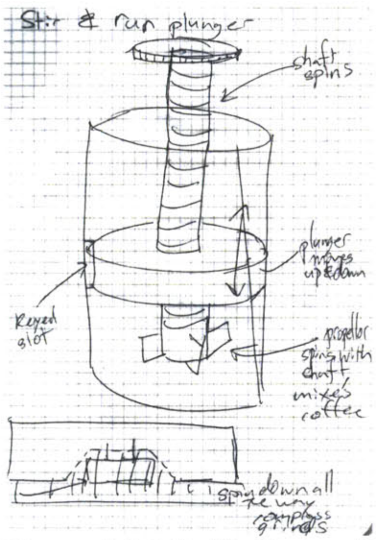
Figure 5.1.5: Blossom machine with labels of selected parts.

5.2 The Support Structure

There are four supports for the system, each made out of a 36 inch piece of 1" square aluminum unistrut, or 80/20. The pieces are connected to a base made from black Delrin plastic using 2 inch long ¼-20 screws. The support structure can be seen in Figure 5.1.2, 5.1.3, and 5.1.4. Unistrut was chosen because it provides a versatile mounting surface for a variety of mechanisms that can be easily reconfigured for additional features to be added as needed. Some extra height was intentionally left at the top of the unistrut in case it proves to be necessary for improvements. The base was given rounded corners and beveled edges for aesthetics (shown in Figure 5.1.4) using a woodworking mill.

5.3 The Plunger Mechanism

The plunger mechanism actuates the plunger assembly, which forces the separation of coffee grounds and the drink. Three actuation mechanisms were explored during brainstorming for this part. It is instructive to consider these preliminary designs as an investigation of other strategies to perform same function, to consider their strengths and weaknesses and better understand the rationale for the final design. The first design that was considered is shown in Figure 5.3.1 and utilized a screw drive with movable plunger inside the cylinder. A benefit of this design was that it allowed for stirring of coffee, simply by including an agitator at the bottom of the screw then rotating it back and forth during the brew. It also could provide the desired function in a compact package. It was eliminated due to the challenge of creating an acceptable seal between the plunger and threaded shaft, as well as challenge of assembly and fabrication. The next design that was considered (shown in Figure 5.3.2) was a belt driven plunger, with a T shaped support structure so that the plunger could move the required distance. The belt drive could provide smooth actuation and flexibility of mounting the drive motor, but added complexity and was eliminated. Finally, an internal combustion engine crankshaft, rod, and piston setup was considered, as shown in Figure 5.3.3. This design was very attractive because of the mechanical advantage that the piston gets at the lowest point of its compression, due to the positioning of the crankshaft and rod. This mechanical advantage could allow the machine to crush coffee beans by mashing them, and would allow the usage of a lower power motor for general usage. However, once again it was determined that the assembly of such a system would be too complex for the purposes of this initial test piece. All of these designs provide advantages that will be kept in mind during future innovation.



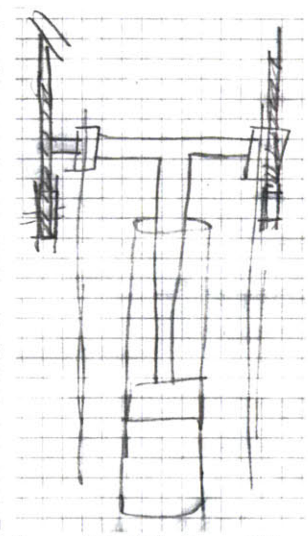
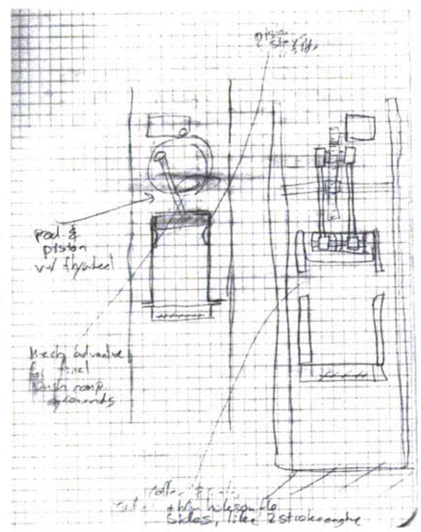
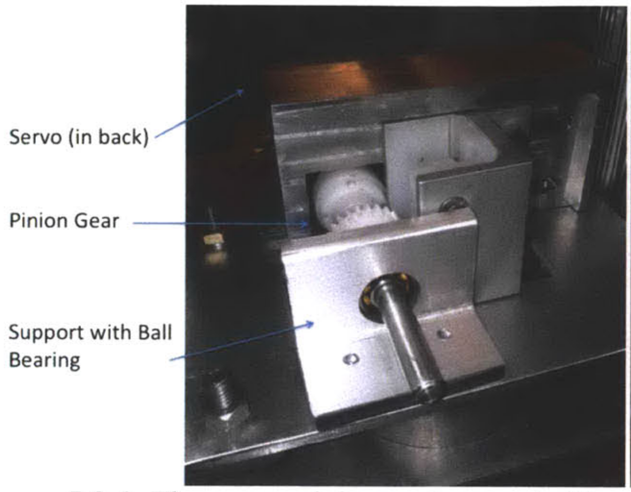



Figure 5.3.1 (left): Rotary Drive with Stirring. **Figure 5.3.2 (middle):** Belt Drive.
Figure 5.3.3 (right): Internal combustion engine style crank and connecting rod drive.

The final design consists of a rack and pinion system and operates with a servo that turns the pinion gear, which causes the linear actuation of the piston. As shown in Figure 5.3.4 and 5.3.6, the pinion gear is press fit on to an aluminum piece of hex stock, which is supported by the servo on one end and is supported by a ball bearing on the other. The servo is attached to an aluminum support block and the rack is attached to a piece of unistrut that rides in an aluminum plain bearing guide block, which is also attached to the support block. The whole assembly is capable of maintaining contact between the servo, rack, and pinion independent of attachment to a support structure. The assembly is attached to a slotted support plate, which allows for the assembly to slide side to side to ensure that the plunger is centered in the brew chamber. The HiTec HS-805BB mega quarter scale servo is capable of outputting 275 oz-in of torque. With the 1-inch diameter of the pinion, this corresponds to 550 oz or 34 lbs of force.

The first rack and pinion that was chosen had a 1" x 1" x 24" rack that was made of steel, as shown in Figure 5.3.5. It was originally chosen to allow for a 1" long contact between the pinion gear and rack, but this amount of contact proved unnecessary. This rack was far too heavy duty for the application, and was replaced with a nylon rack and pinion, also shown in Figure 5.3.5. The support block was originally designed with slots to connect to the plain bearing guide block, to allow for changes in pinion gear size. However, the original rack and pinion was so different in dimensions than the final that a new hole had to be drilled in the support block to reposition the plain bearing guide block accordingly. Testing also showed that the driveshaft needed to be supported on both ends, or else the pinion gear would come out of contact with the rack when under high torque. An aluminum bracket was cut from L shaped stock and fitted with a ball bearing to support the driveshaft, as shown in Figure 5.3.5. Slots were added to the aluminum plate to allow for the bracket to move with the rest of the assembly.



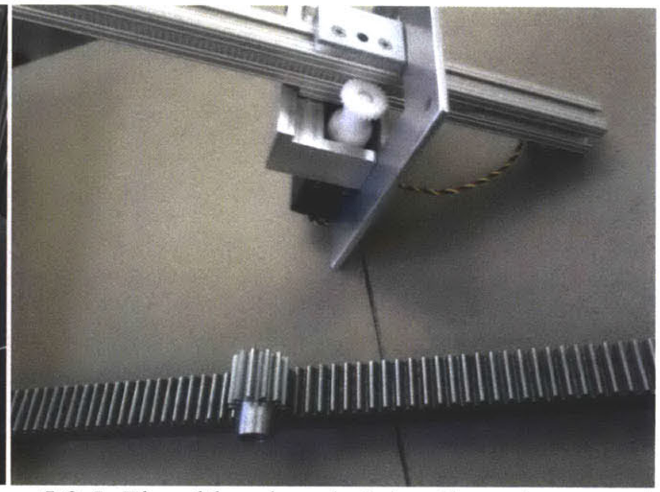


Figure 5.3.4: The servo, pinion, and support. Figure 5.3.5: The old rack and pinion (lower) next to the new rack and pinion assembly.

The support block was machined from aluminum; a technical drawing is included in Appendix A. A Hitec HS-805BB servo was ordered from Hobby Shack and was modified for continuous rotation by removing a plastic rib and disabling the potentiometer. The pinion gear was modified to have a hexagonal center profile, to fit a piece of aluminum Hex stock. The pinion was connected through the servo using two intermediary nylon pieces, as shown in Figure 5.3.6. Mating directly to the servo was a splined piece that was included in the set and had pre-drilled holes for screws. It was turned down on a lathe to a diameter just large enough to still contain the innermost mounting holes. Between the splined piece and the pinion was an intermediary piece of nylon that had two main diameters: one that connected to the splined piece's and one that was the same as the inner diameter of the pinion gear. The intermediary piece was attached to the splined piece using machine screws and was made using a lathe and a drill press. The intermediary piece, like the pinion gear, had a hexagonal hole cut into it and had the hexagonal axle press fit in.

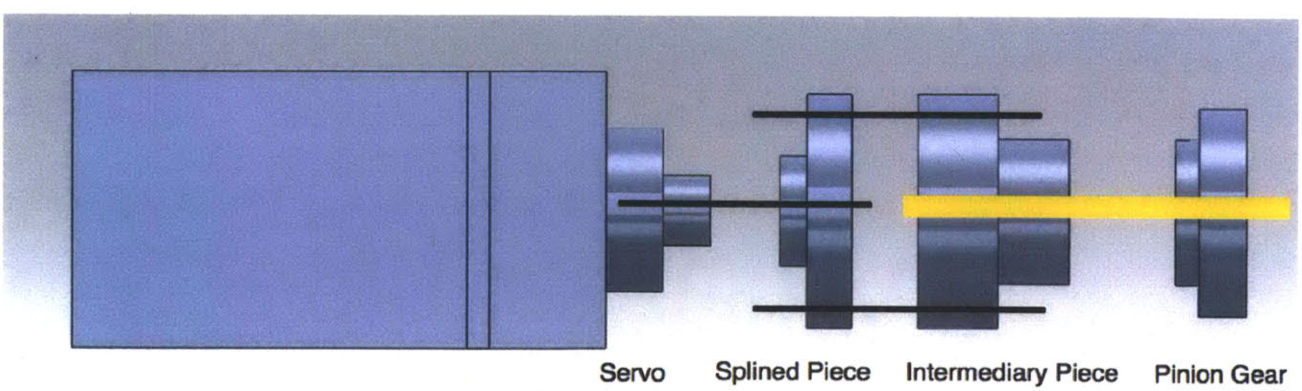


Figure 5.3.6: Servo, Splined Piece, Intermediary Piece, and Pinion Gear. The horizontal black lines refer to the location of screws, the yellow box shows the location of the hex piece.

5.4 The Brew Chamber

The brew chamber is a stainless steel tube in the center of the device that holds the coffee and water during brewing. As can be seen in Figure 5.1.4, a heating element is wrapped around the brew chamber to regulate the temperature of the brew chamber. Some early designs were also based on an acrylic brew chamber, but it quickly became apparent that high-grade stainless steel was better suited to this project due to its high thermal conductivity and chemically inert nature. Wall thickness was minimized to reduce the thermal mass the heating element would act upon, allowing more rapid control of temperature. The final design has a 1/16" thick wall.

The filter assembly is attached to the brew chamber with a hinge. The upper part of the hinge will be described in this section, as it is welded to the brew chamber. The upper hinge started as a solid block in Solidworks, as shown in Figure 5.1.2. An early plan was to make the part out of a solid piece of stainless steel and weld it to the stainless steel brew chamber, but this plan was scrapped due to the high cost and weight of that much stainless steel, as well as the time-intensive machining process that would have been required. The design was improved by replacing the solid steel block with a hollow piece created by welding together four steel plates (shown in Figure 5.4.1). When this box was welded to the tube, however, the welding caused the tube to shrink and created a point where the piston would jam. To fix this problem, the tube was honed to wear off some material, and the piston was reduced in diameter by 0.004".

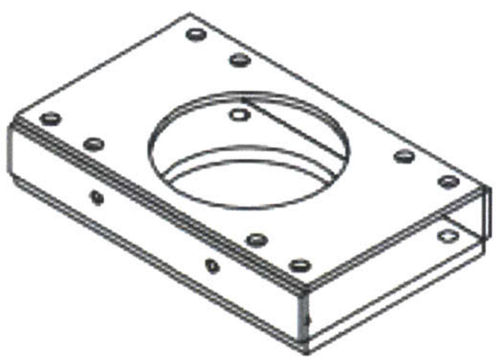


Figure 5.4.1: Four Part Welded Upper Hinge

5.5 The Filter Mechanism

The filter mechanism is responsible for separating the coffee from the water and is located at the bottom of the brew chamber. The Aeropress has a similar piston-in-cylinder structure, with a filter at the bottom. When using the Aeropress, the coffee grounds are compressed against the filter and form a compressed puck that can be easily ejected with minimal loose grounds. The filter piece in this machine was designed to accomplish a similar effect. Early designs focused on a hinged mechanism that could swing the filter out of the way to allow for the ejection of grounds, as shown in Figure 5.5.1. The design shown in Figure 5.5.2 had a three part stackup, with a square filter and intermediary piece to seal against the brew chamber. The square filter even made it into some early computer models, as show in Figure 5.5.3. However, concerns that such a filter shape could allow for leaking caused a redesign with a round filter.

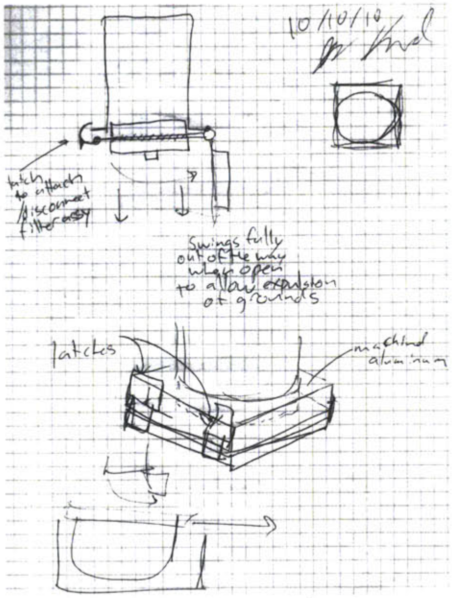


Figure 5.5.1: Filter with Hinge

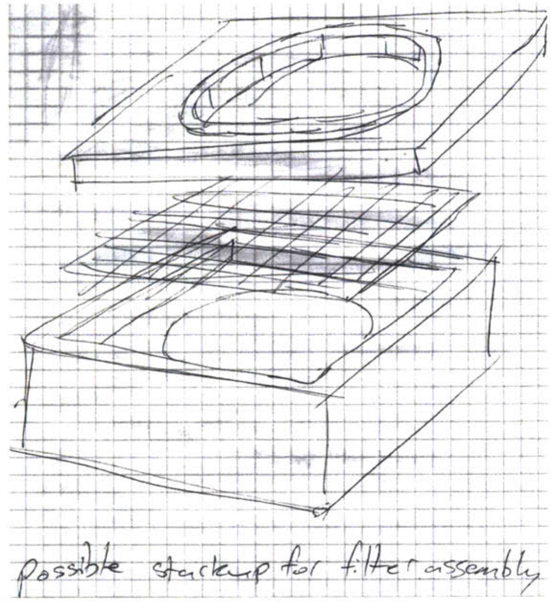


Figure 5.5.2: Early Filter stackup

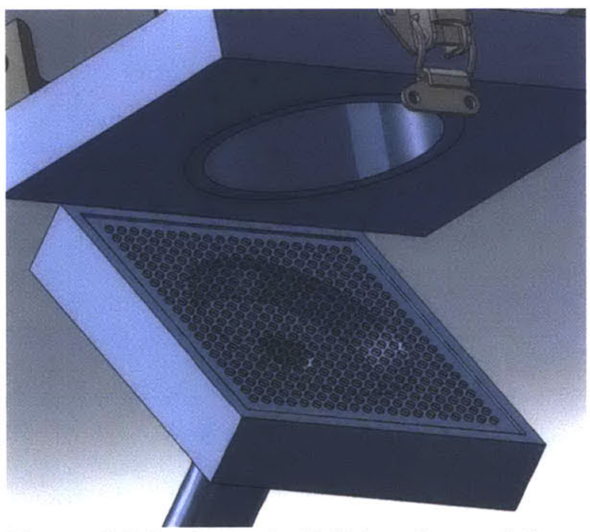


Figure 5.5.3: An early Solidworks model

The final part was made of black Delrin and has a small volume below the filter to collect the coffee and funnel it into the coffee cup. The filter support is a circular metal screen cut to fit the hinge piece, upon which the user places paper coffee filters. The hinge latches in the front to close tightly on an o-ring, which is inserted in the groove around the filter support and attached with adhesive (Elmer's glue-all was used). Figure 5.5.4 shows these pieces in an exploded view. After brewing coffee, the plunger compresses the grounds into the filter assembly, which swings out of the way to allow the grounds to be ejected. A detachable hinge was used to provide for easier disassembly. Figure 5.5.5 shows the profile of the lower hinge with the features labeled.

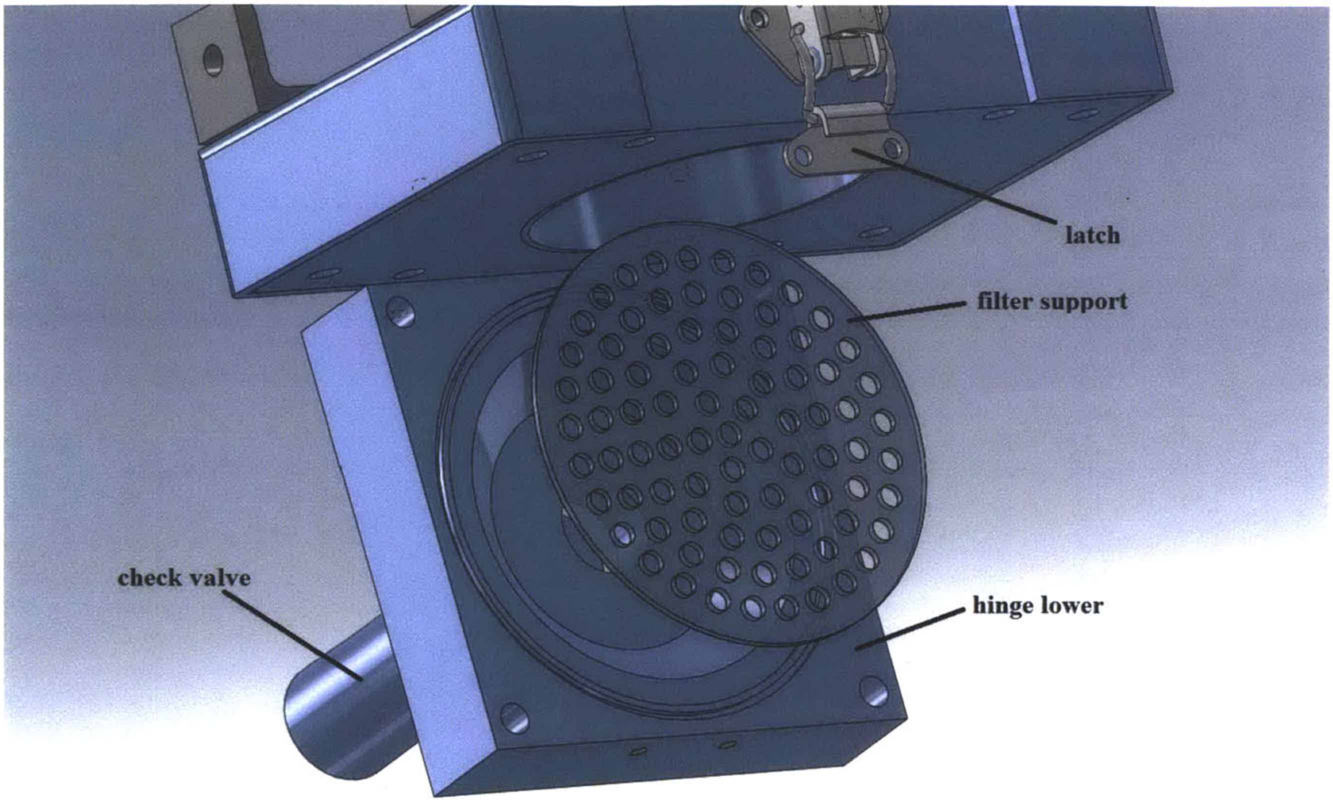


Figure 5.5.4: Final Lower Hinge Assembly

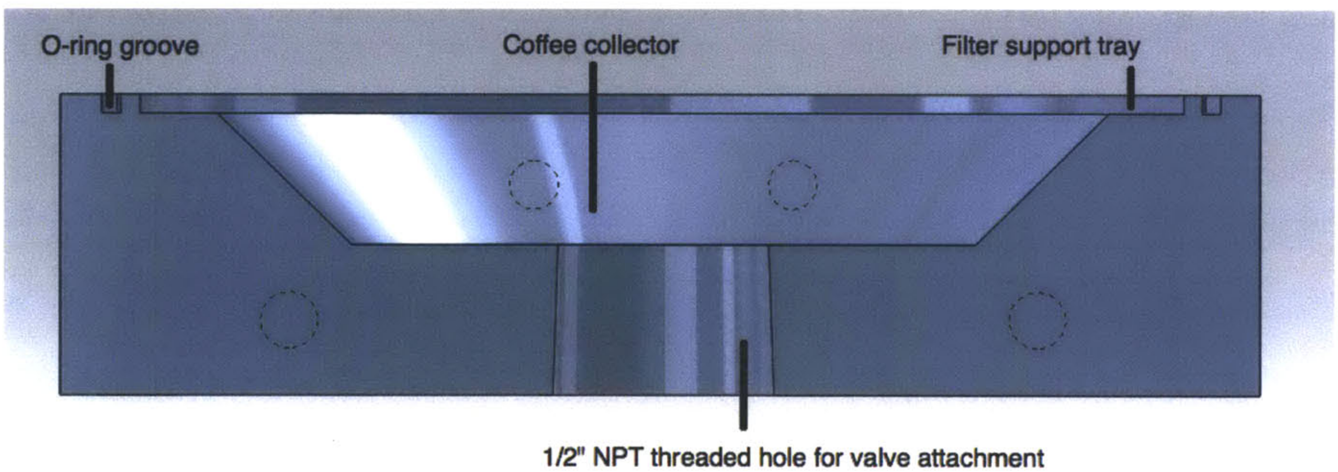


Figure 5.5.5: Final lower hinge section view

5.6 Control Electronics

The control electronics execute two functions: temperature control and servo control. These two functions are the key for the system to execute control of brew temperature and duration. An Arduino programmable microcontroller is utilized to perform these functions. The Arduino is a suitable choice because its open source community provides ample resources for learning, and its flexible nature allows it to implement future innovations.

A HiTec HS-805BB servo was used to actuate the rack and pinion. A flexible silicone-rubber adhesive backed heating pad with $10W/in^2$ power was used to heat the brew chamber. The power to the heating pad was regulated by an Omron G5LE relay. Temperature information was sent to the

Arduino from a K-type thermocouple and AD595 thermocouple amplifier. Figure 5.6.1 shows the electrical connections between these components. Appendix B contains the code that was used to operate the machine.

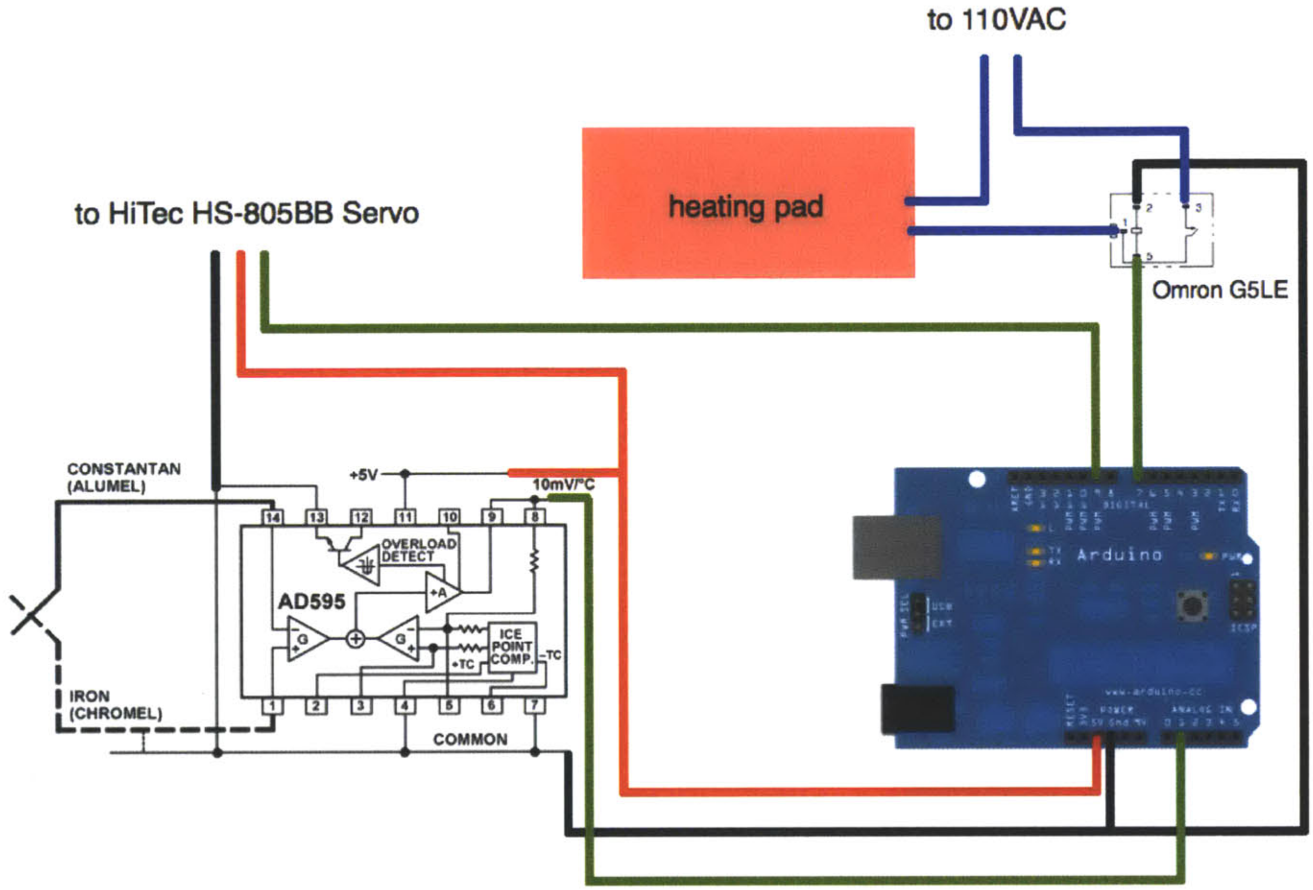


Figure 5.6.1: Electrical schematic showing the connections between the Arduino, the relay, the servo, the thermocouple, and the AD595 amplifier.

6.0 Testing

After design and fabrication, the machine was tested to verify its performance against the manual testing conducted in Section 3. This testing allowed verification of the machine's performance in a variety of situations, and the chance to uncover its strengths, weaknesses, and areas for improvement. Coffee was brewed in the machine at three different temperatures for five different durations of time.

6.1 Test Equipment

Testing of the machine was performed with the same materials and test equipment as the testing performed in Section 3 so that the results could be compared. Folger's regular grind medium roast coffee was mixed with distilled water. The pH sensor that was used requires at least 10 μS conductivity to obtain an accurate measurement. The distilled water was measured to have a 1.3 μS conductivity, which made it impossible to determine the initial pH and makes absolute comparison of pH with the previous results problematic, although trends of pH as a function of temperature can still be observed. The same HM Digital pH and TDS sensors were used, as well as the same Vernier temperature probe. A single-input Vernier Labpro interface was used instead of the four-input interface used previously, and was connected to the same computer running Vernier Logger Pro. Temperature was measured using two different sensors: the aforementioned temperature probe above the thermocouple connected to the Arduino. The temperature probe was used to verify the measurements taken by the Arduino and log the temperature during the brew.

During initial testing, it became apparent that the machine required at least 500 mL of water to be able to brew coffee without overheating. When filled with less than that volume, the water did not fill the brew chamber enough to cover all the parts being directly heated by the heating element, and the system became so hot that it started smoking. Therefore, verification testing was performed with 500 mL of water and 62.5 grams of Folger's coffee, five times the amount used previously.

6.2 Test Methods

First, circular filters were cut from the filter paper. To cut the filters, the metal screen was placed on the filter paper and a line was traced around the outside with a pencil. A stack of filters was cut simultaneously with scissors. The filter was pre-wet to help it to adhere to the metal screen and both were inserted into the lower hinge. The lower hinge was then closed, and the ground coffee was added through the hole in the front. The distilled water was measured to 500mL in a plastic measuring cup and was heated to boiling in a microwave. After heating, the cup was removed from the microwave and the water temperature was monitored with the computer and temperature probe. Once the temperature dropped to the desired point for the test, it was poured into the machine through the hole in the front. The temperature probe was then inserted into the brew chamber and used to stir the mixture for the first 10 seconds of brewing. During some brews, the change in temperature over time was recorded to verify the performance of the system at maintaining a constant temperature. The duration of brew was programmed in the computer and the Arduino actuated the plunger after this period of time and pushed the coffee through the filter, out the valve and into the cup. The coffee was then measured for pH and TDS, and was also tasted.

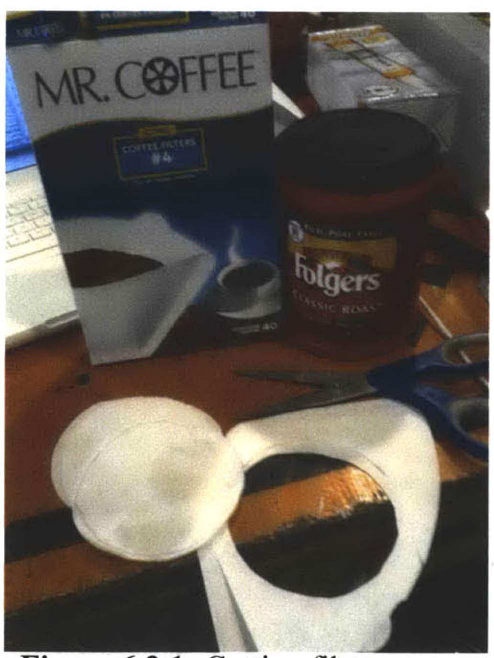


Figure 6.2.1: Cutting filters

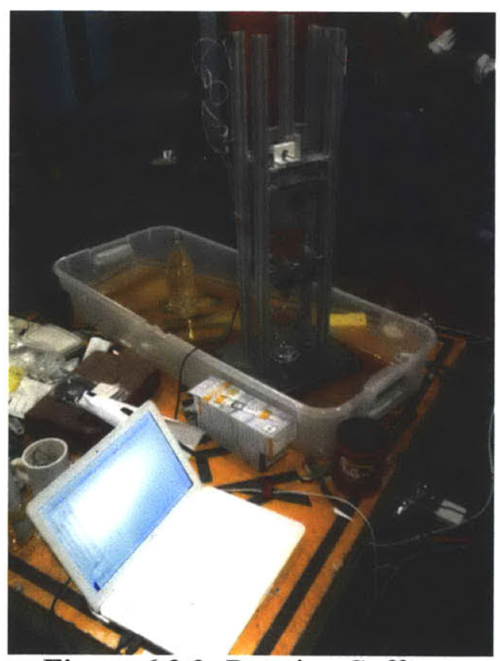


Figure 6.2.2: Brewing Coffee

6.3 Analysis of Blossom Temperature Regulation

The Blossom was designed to keep brew temperature constant to aid in experimentation with coffee flavor extraction. Results were recorded from a thermocouple that reported temperatures to the machine, which were verified by a temperature probe connected to the computer and recorded with Logger Pro software. The performance of the Blossom with maintaining a set temperature was found to be inconsistent, as can be seen in Figures 6.3.1, 6.3.2, and 6.3.3. However, the machine's recorded temperature, shown in Figure 6.3.1, was consistent with maintaining a constant brew temperature. This discrepancy between machine recorded temperature and externally reported temperature indicates that one or both of the temperature recording devices was in error. It is likely that the Blossom machine was in error, as the thermal probe with Logger Pro has been used in the past with consistent results. Also, the thermocouple underwent some damage during usage. It was threaded through the plunger and compressed into the coffee grounds when the brew was pushed through the filter. This arrangement caused significant wear and tear on the casing for the thermocouple wires, which had to be repaired several times with electrical tape. Thermocouples operate by measuring the voltage between two different metals at a junction.[19] They can fail by forming a connection along the two wires at a location other than the temperature to be measured, thus the wear that was being experienced by the thermocouple wire casing was of concern. By the sixth and seventh tests, the thermocouple had to be carefully positioned to avoid forming a connection between the wires and recording the temperature at the wrong point. This wear on the thermocouple could have caused incorrect temperature reporting.

The temperature recorded by the Blossom is shown in Figure 6.4.5.

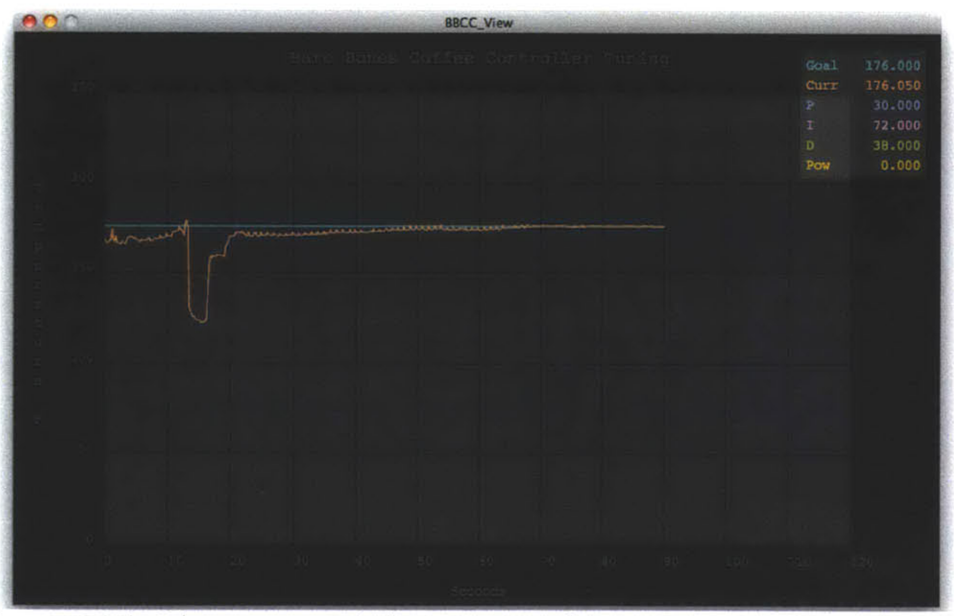
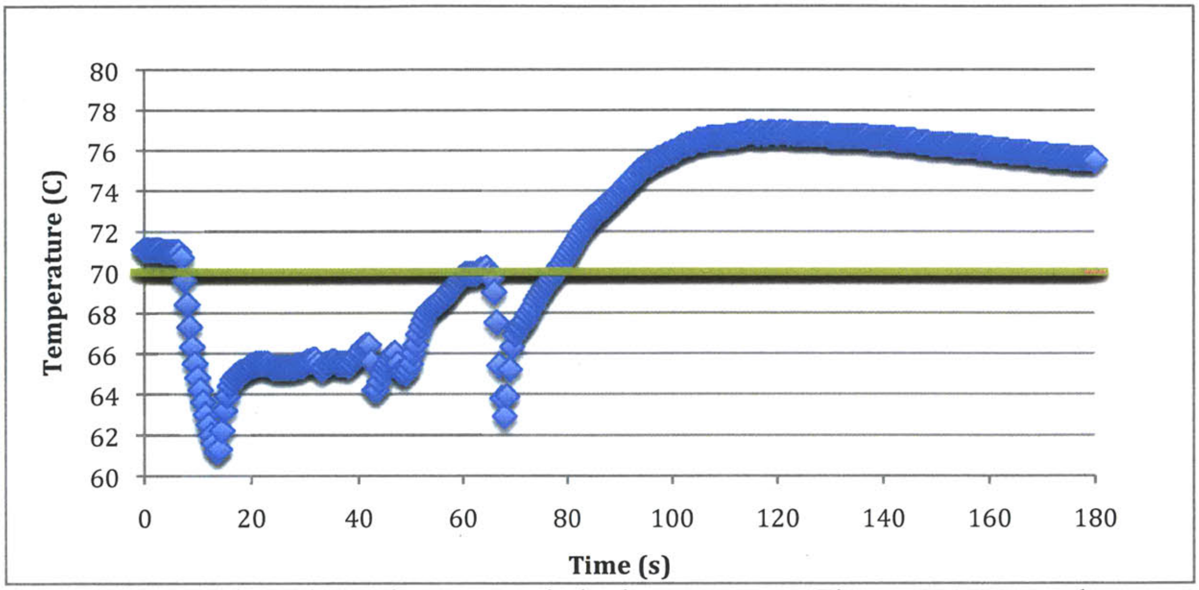


Figure 6.3.1: Blossom recorded temperature for a 70C 3 minute brew (the same brew depicted in Figure 6.3.2), the machine recorded that it was at the proper temperature when it actually overheated the brew, as can be seen in Graph 6.3.2. This situation was likely due to a faulty thermocouple. The seconds in the X-axis do not correspond to seconds of brew time.



Graph 6.3.2: 3 minute brew time at 70C desired temperature. The temperature probe was preheated in the water (0-10s). It was then mixed with the ground coffee, which caused the temperature drop (10-40s). At 60s, it was touching the wall of the brew chamber and was moved to the center of the brew, which rose in temperature from 80-100s and cooled from 100-180s.

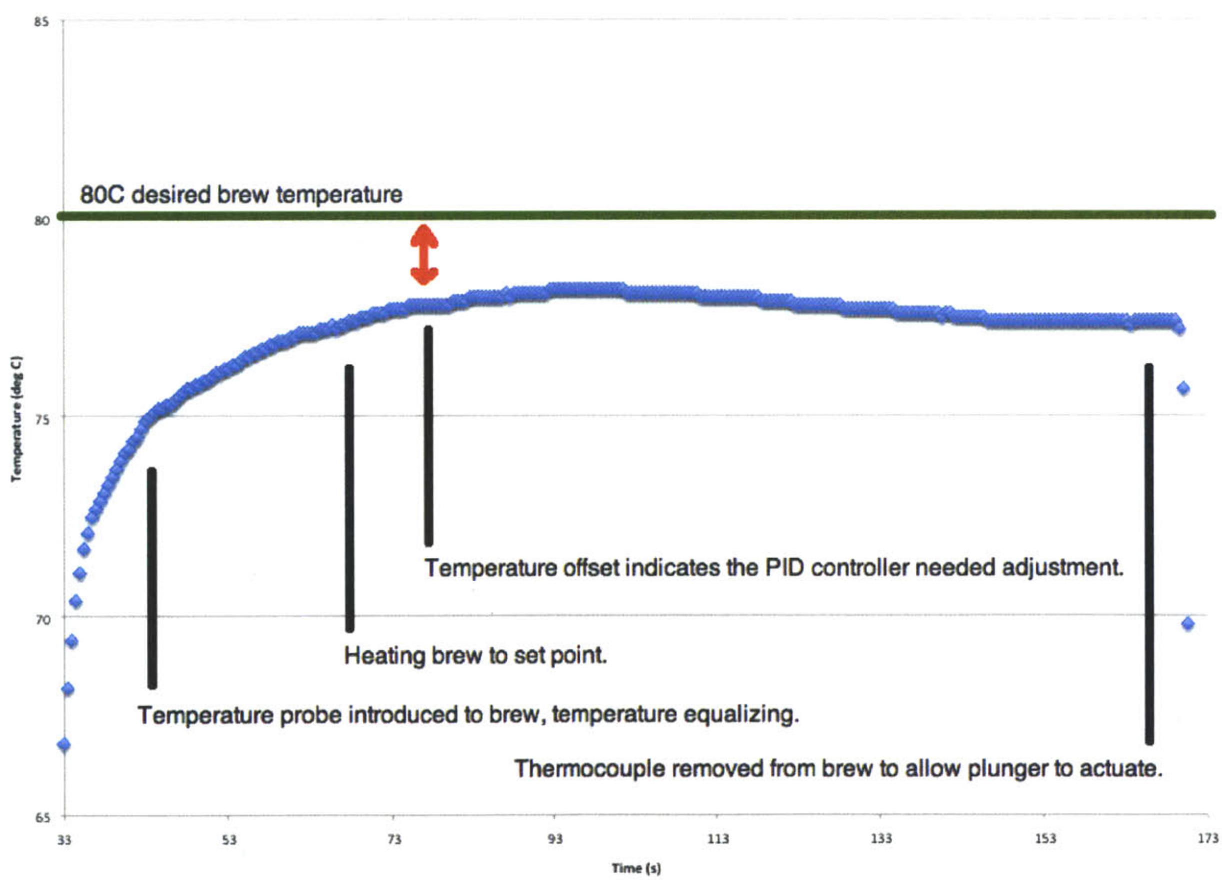


Figure 6.3.3: Blossom performance with 80C set point. ~2.5C temperature offset indicates the PID controller needed calibration, or that the thermocouple was reporting a different temperature than was recorded by the sensor that created the results displayed here.

6.4 Test Results and Discussion

Seven tests were performed. Five tests were performed at 90 Celsius for durations of 1, 2, 3, 4, and 5 minutes. Two tests were performed for a 3 minute duration at 80 and 70 C. These test points allow examination of both time and temperature dependencies and the results are displayed in Table 6.4.1. The error range for the pH tester is +/- 0.02, so the overall measured difference of 0.05 is only 0.01 outside of the confidence interval of the test device.

Table 6.4.1: Experimental Data. The conditions with the best flavor were at 80C with a 3minute brew duration. This data point was within the range of conditions in Section 3 found to have optimal flavor.

Temperature (deg C)	Duration (min)	pH	TDS (442 ppm)
90	5	5.63	2510
90	4	5.68	3300
90	3	5.57	2210
90	2	5.61	2750
90	1	5.55	1800
80	3	5.63	1540
70	3	5.65	1770

There was a correlation found between temperature and pH, as shown in Figure 6.4.1. This conclusion corresponds with the data found in Section 3, where higher temperatures correspond to lower pH. The total difference in pH displayed in Figure 6.4.1 is 1.5%. It was also noted that the cup of coffee brewed at 80C for 3 minutes had exceptional flavor, far superior to the cup brewed at 90C and better than the one brewed at 70C.

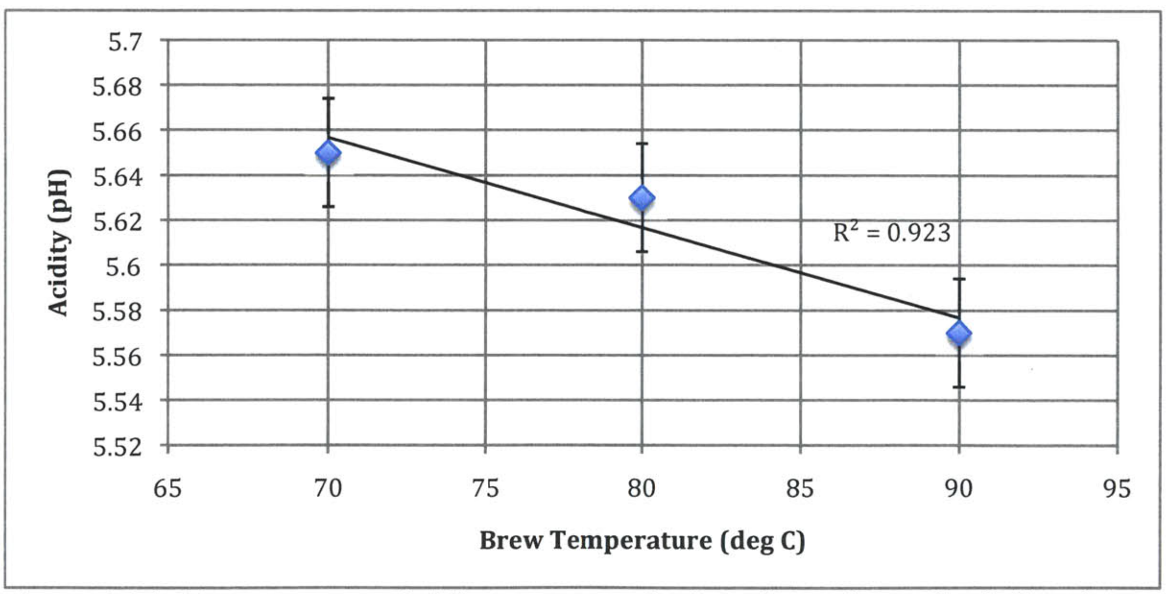


Figure 6.4.1: Temperature vs Acidity at constant 3 minute brew time. The fit line shows a correlation between brew temperature and acidity of coffee. Error bars correspond to the specified precision of the pH sensor.

As shown in Figure 6.4.2, no correlation was found between brew temperature and TDS. This is consistent with what was found in Section 3, where TDS corresponded to a certain regions of temperature and duration, as opposed to either variable independently.

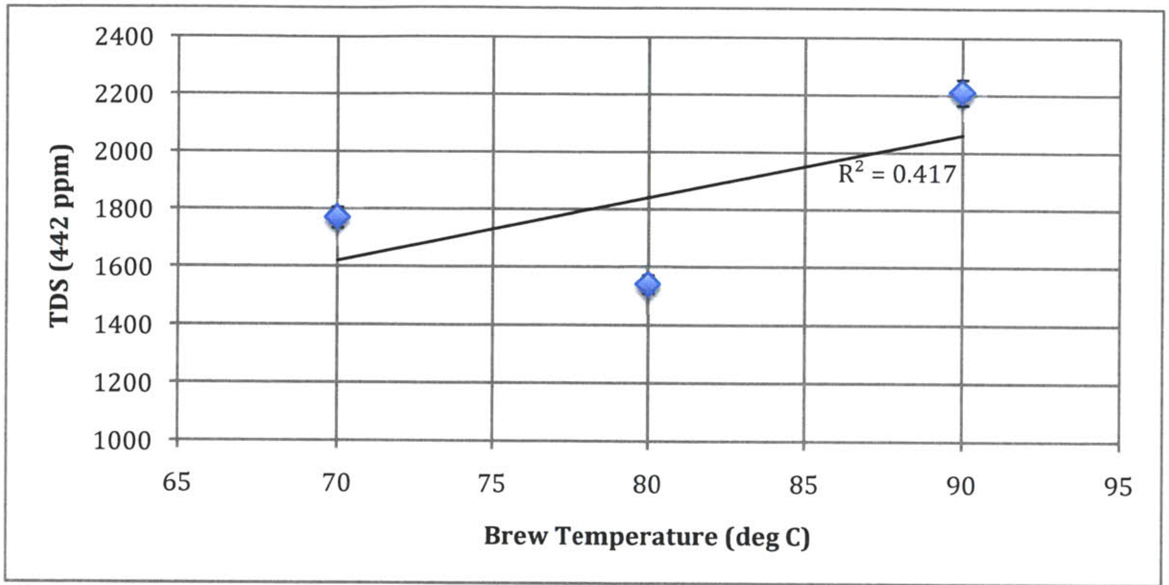


Figure 6.4.2: Temperature vs TDS at a constant 3 minute brew time. The fit line does not show a correlation between brew temperature and TDS.

Analysis of pH values brewed at 90C for varying durations did not reveal a clear relationship to previous testing. Prior testing showed that pH stays relatively constant when brewed at 91C for varying durations between 1 and 5 mintues, with a change in pH of 0.05 between the highest and lowest recorded values. In this test the change in pH of 0.13 was measured, more than twice as much as previously. Furthermore the pH values in this test had an average of 5.61, which is 10% higher than the previous average of 5.08. The difference in average pH was 0.53, which may have resulted from differences in initial pH of the distilled water. As discussed in Section 6.1, measurement of initial pH of the distilled water was problematic. For this reason, Blossom results in Figure 6.4.3 are shown in their original range, as well as adjusted by the average difference in pH.

Although the ratio of coffee to water was the same as previously, the volume of each was five times greater. This increased volume could have caused the coffee to dissolve differently into the water and increased the importance of agitation during the brew because of the ground coffee forming clumps. It was noticed during testing that the larger volume of coffee seemed to respond differently to being mixed with water, but it was difficult to observe in what way due to the shape of the brew chamber. However, there seemed to be more clumping of the coffee grounds so that their exposure to water was less uniform than in the previous test. This clumping could cause inconsistent results.

As stated in Section 6.1, the measurement of initial pH of the distilled water was problematic. The Blossom results are shown being adjusted down by the difference in average pH (0.53) to account for possible differences in initial water quality.

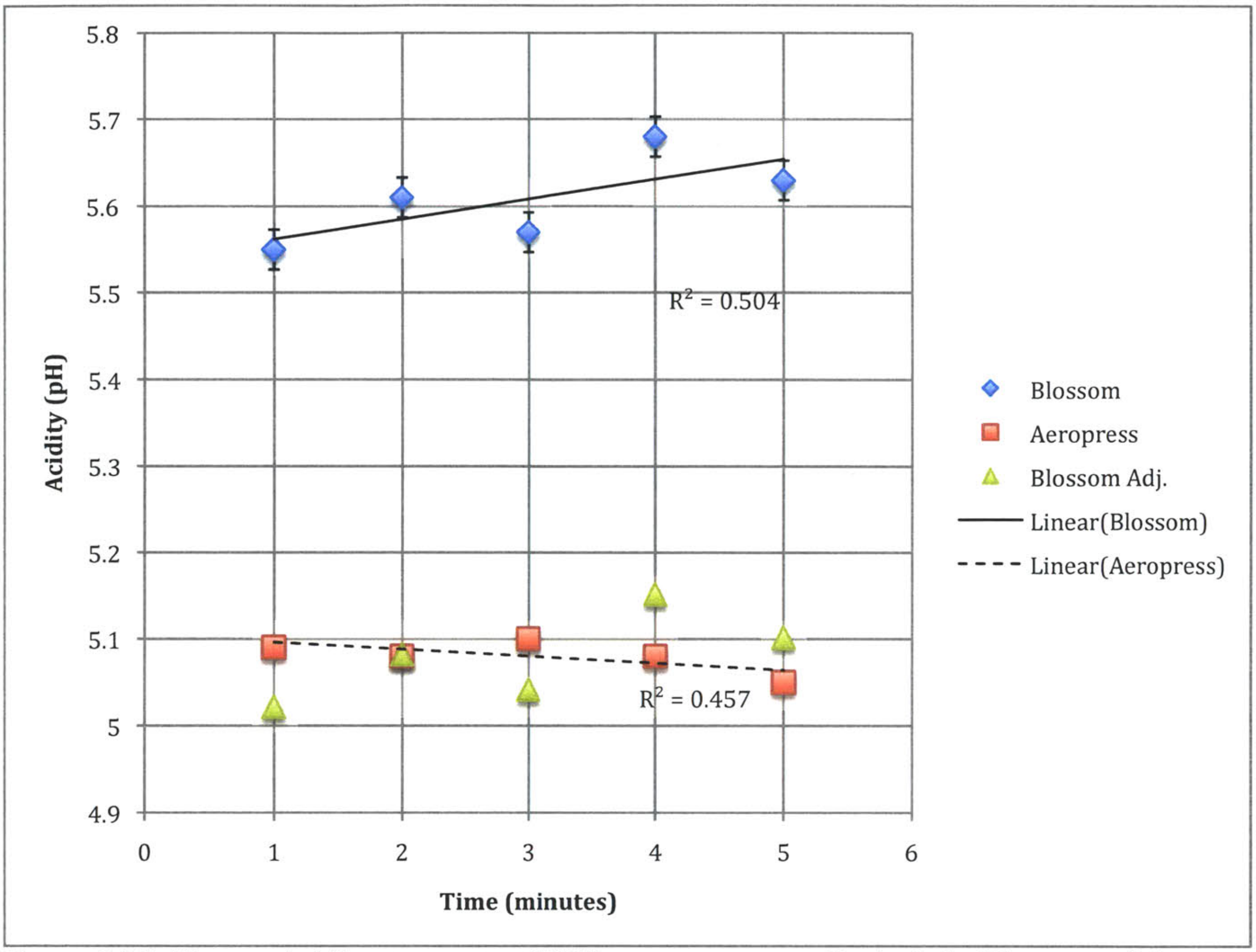


Figure 6.4.3: Time vs Acidity at a constant 90C brew temperature. For both the Blossom and the Aeropress, optimal flavor was found at 3 minutes brew duration.

Analysis of TDS as a function of time showed the Blossom machine producing coffee with TDS values closer to the original results, which are graphed in Figure 6.4.4. Both examples showed a general trend of TDS increasing with brew duration. The Aeropress produced coffee with maximum TDS at 3 minutes brew time, whereas the Blossom produced coffee with the second lowest TDS at 3 minutes duration. This difference is notable because when testing the Aeropress, the best flavors were noted at 3 minutes brew time and a peak in TDS was found at that brew duration as well. The aforementioned mixing issues in the Blossom machine could have contributed to this difference in results.

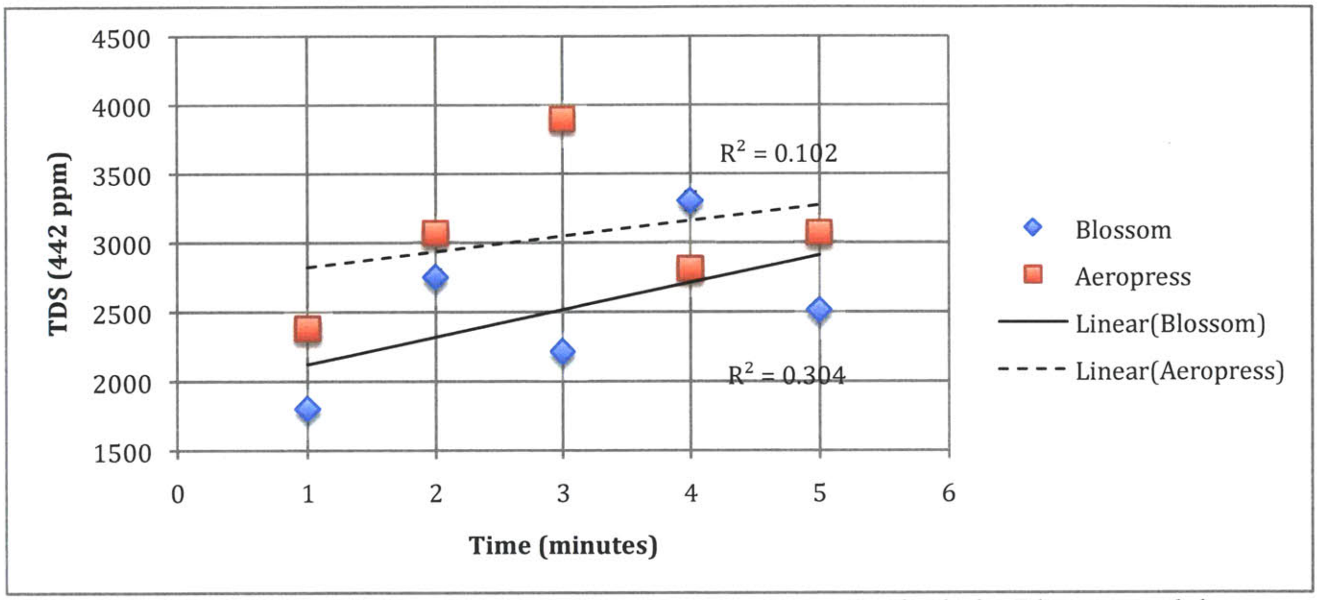


Figure 6.4.4: Time vs TDS at a constant 90C brew temperature. For both the Blossom and the Aeropress, optimal flavor was found at 3 minutes brew duration.

7.0 Conclusion

A coffee machine capable of controlling brew temperature and brew time was designed, manufactured, and tested. A user study was conducted, and an internet-connected coffee brewing system was designed. A bench-top prototype machine was fabricated and tested by making seven cups of coffee at different temperatures and durations, which were measured for acidity and total dissolved solids. The values were compared to earlier testing that had been completed with manual methods. The acidity of the coffee made with the automatic system followed a similar trend to the manual methods. Two of the cups of coffee brewed with the machine showed exceptional flavor, indicating that the Blossom machine has the potential to produce excellent coffee. Future work is planned to incorporate an agitator, and to modify the plunger mechanism such that the machine could pressurize the brew before it ejects it. There are also plans to replace the heating element with a Peltier device, and create a control system that can allow the user to select a custom temperature profile, with both heating and cooling coming from the Peltier device. The future control system should also allow dynamic modulation of pressure and stirring during the brew, and it may even be possible to develop an open loop system that automatically maximizes TDS levels.

8.0 Acknowledgements

Dr. Barbara Hughey

Dr. Mostafa Hamed

Caroline Bell

Chris Timbrell

James Park Brannon

Abraço, Intelligentsia, Oslo, Blue Bottle Coffee, Café Grumpy

Rachel Besler and Thomas Eskew (interviewed at Abraço)

John Kuempel Sr.

John Kuempel Jr.

Eileen Kuempel

Joshua Kuempel

Pappalardo Laboratory; Dick, Joe, Jim, Steve, Bill.

Diecraft Machining and Engineering in Cincinnati; Skip and James.

Central Machine Shop at MIT; Andy and Peter.

Bryan Maycomber

John Romanishin

Blair Gagnon

Gary Scheffield

Chuk Werley

Jessica Saniehoff

The Brothers of Kappa Sigma, Gamma Pi

9.0 References

[1]"Coffee Flavor- The Taste Characteristics of Coffee". Gourmet-Coffee-Zone-Online. Ed. Mark Harris. Gourmet Coffee Society. December 2010 <http://gourmet-coffee-zone.com/coffee-flavor.html>

[2]"Blossoming". illy.com. © 2008. illy. January 2011. <http://www.illy.com/wps/wcm/connect/us/illy/the-world-of-coffee/origin-and-blend/blossoming/>

[3]Coffeeresearch.org. Ed. Michael Griffin. © 2001-2006. Coffee Research Institute. December 2010. <http://www.coffeeresearch.org/agriculture/coffeeplant.htm>

[4]"The Home Roasting Tradition". Sweetmarias.com. Sweet Maria's Coffee. January 2011. <http://www.sweetmarias.com/tradition.php>

[5]Schultz, Howard. Pour Your Heart Into It. New York: Hyperion, 1997.

[6]Brain, Marshall and Jessika Toothman. "How Coffee Makers Work". Howstuffworks.com. How Stuff Works. January 2011. <http://home.howstuffworks.com/coffee-maker.htm>

[7]Photo credit to Frank C Müller. January 2011. <http://en.wikipedia.org/wiki/File:Kaffeemaschine_fcm.jpg>

[8]Photo credit to Chemex Corp. January 2011. <http://www.chemexcoffeemaker.com/cm-6a.jpg>

[9]Rancilio Macchine per caffè s.p.a. - Viale della Repubblica 40 - 20010 Villastanza di Parabiago - Milano Italy tel +39 0331 408200 - fax +39 0331 551437 - info@rancilio.it

[10]Photo credit to Green Mountain Coffee Roasters Inc. January 2011 <http://www.keurig.com/brewers/b140-brewering-system>

[11]"Why does the Aerobie Aeropress Coffee Maker Brew a Richer, Smoother Cup of Coffee or Espresso?" Aerobie, Inc. January 2011. < http://www.aerobie.com/Products/aeropress_story.htm >

[12]Honan, Mathew. "The Coffee Fix: Can the $11,000 Clover Machine Save Starbucks?" Wire Magazine Online. 21 July 2008. December 2011. <http://www.wired.com/gadgets/miscellaneous/magazine/16-08/mf_clover>

[13]James Park Brannon, Café Grumpy barista. Private Communication. 16 October 2010.

[14]Caroline Bell and Chris Timbrell, Café Grumpy co-owners and co-founders. Private Communication. 16 October 2010.

[15]Luminairecoffee.com. Luminaire Coffee Company. December 2010. <http://luminairecoffee.com/>

[16]Photo Credit to How Stuff Works. January 2011. <http://static.howstuffworks.com/gif/clover-coffee-maker-7.png>

[17]"The Bunn Trifecta Experience" © 2010 Bunn-O-Matic Corporation.

<http://www.trifectaexperience.com/tech_enabled.html>

[18]HM Digital COM-100 EC/TDS/TEMP Combo Meter User's Guide, HM Digital Inc.

[19]"Acid Rain Experiments – Experiment 1 – Measuring pH". EPA.gov. 8 June 2007. Environmental Protection Agency. January 2011. <http://www.epa.gov/acidrain/education/experiment1.html>

[20]"Introduction to Thermocouples". Omega.com. Omega Engineering, Inc. January 2011. <http://www.omega.com/techref/themointro.html>

Appendix A: Additional Design Thinking

The ideal machine for the home would not only create a superior coffee drinking experience, but would also making waking up a better experience. When people normally drink coffee in the morning, they are in an incredibly fragile and personal state, a situation that even many of their closest friends never see them in. They just woke up and may not be thinking clearly yet. Their hair is messy, they need to shower, and they might even be late for work. The perfect coffee machine would understand this use case scenario, and be comfortable and easy to use. It would brew them a cup of coffee so delicious and energizing that the customer couldn't help but walk out of their house in the morning with a smile on their face and jump in their step.

The perfect home brewer would also be equipped with a touchscreen and internet connectivity. The internet connectivity should be both wireless and allow itself to be plugged in, and should be incredibly easy to connect and set up. The brewer would automatically upload the brew parameters to a centralized database to store information about how to brew coffee. Each bean and each different roast of each bean has unique properties, so initially it could be a challenge for users to find the correct brew parameters to make themselves superior cups of coffee. This situation must be avoided; the perfect home brewer should be able to make a superior cup of coffee the first time and every time it is used so that the customers can learn to trust it. By incorporating internet connectivity, users could download the perfect parameters for whatever beans they happen to be using. They could also rate the cups they've made and create a community of coffee enthusiasts who research brew parameters for beans.

Incorporating a touchscreen and internet connectivity on the home coffee machine provides for other interesting possibilities. It allows for the sale of coffee beans and other products directly from the machine. If a customer were having trouble finding high-end gourmet beans, they could access an online store directly from their home machine and find well-reviewed beans. Because the machine is able to fully control brew parameters and consistently brew the same cup of coffee, internet connectivity would allow users to "gift" cups of coffee to their friends. For a nominal fee, they could have a single serving of fresh beans delivered to a friend, with parameters that are automatically uploaded to the machine, allowing them to literally share a unique cup of coffee. This presents a revenue stream for the company, allowing the company to continue to profit from a machine after it was sold. Furthermore, the machine could provide smart recommendations to consumers based on other beans they've tried, like some websites are starting to do with online retailing. The machine would not need to sell only coffee. Directed advertising would also be a possibility, and sale of other kitchen related products is an adjacent market. If a touchscreen, internet enabled coffee machine were to become popular and find its way into homes, it would be another leap in development of web technologies, providing internet usage in an environment that is not a computer or smartphone.

The touchscreen and internet connectivity also provides a platform for educating the customer about the machine and coffee. The machine could display information about the bean being brewed, such as where it was grown. It could display the current weather of the place the bean was grown, and show pictures or even a live webcam from the farm. It could trace the history of that bean from the farm to the machine, and help create a connection between the user and the cup being brewed. If the coffee machine was equipped with its own touchscreen, as opposed to simply utilizing a smartphone application for interaction, the company would be able to better control the user experience and eliminate bugs. The internet connectivity of the machine would allow for smartphone interaction, however. With a smartphone app, the user could interact with the machine when not physically next to it. This way they could leave the machine preloaded to make coffee, then instruct it to start brewing before they ever even reach the machine. They could also check the status of cups they've brewed in the past, and participate in the online community of people who own the machine.

The touchscreen provides the user increased control over the brew. Current coffee machines either allow the coffee to cool naturally while brewing, or brew it at a single temperature. With a touchscreen and an innovative feedback control system, the user could program a desired cooling curve for the brew. For example, they could specify that it should brew at 190F for the first thirty seconds, cool to 180F for twenty seconds, then reheat to 200F for forty seconds. Such interaction could be accomplished by visually dragging desired temperature lines on a graph, then seeing how close the machine comes to actually accomplishing the desired curve. This sort of coffee brewing has not been done in the past, and would provide the user with a truly unique coffee experience.

With coffee machines at home, in the workplace, and in cafés that are capable of being programmed to personalized brew parameters, there is the potential to create an integrated system. The user needs to be able to brew their favorite cup at any of these machines and consistently receive the same, high quality drink. The user experience needs to be congruent and familiar at all three locations.

To truly provide the best possible user experience, the machines would only be used in wholly owned cafés. By owning all of the cafés that the machine is located in, the company owner would be able to guarantee that the machines are working as they are supposed to. The owner could also customize the customer experience in the shops to be an immersive environment that's perfectly suited to the personality of the machine. Furthermore, the cafés could be used as company stores to distribute and service the home systems. The baristas could not only provide customers with brewed coffee, but also educate them on tips and tricks for their home system. The stores could even act as mini coffee schools, with classes and pamphlets to educate customers on the intricacies of coffee.

A possible café design would feel something like a cross between an Apple store and a Starbucks. It would be a Third Place for friends to meet up outside of work and home, with comfortable seating and conversation pieces. The store experience should be decidedly high-end, feeling more classy and modern than Starbucks, sort of like a hip lounge that could be found in New York or Hollywood. There would be high-tech attributes to the stores as well. Because the coffee machines are network connected and can brew the same cups around the world, the stores could be as well. Stores would have live webcams or some other information being fed to them from other stores, so that customers could understand that their experience, although unique to them, is also a part of a greater system. The seating should be incredibly comfortable, more so than anything the customers have experienced before. Modern music and art should be displayed. Blossom should have a smartphone application that expands possibilities for users in the store. For example, users could pay for their coffee using their near-field-communication equipped cell phone. Blossom could also create its own currency like on Second Life, so that customers could pay for things internationally without worrying about exchange rates. Also, with its own currency, possibly called Blossom Points, customers could win points playing in-store games, or by completing challenges. For example, a customer could get free Blossom points for visiting three different stores in less than an hour. Challenges could be proposed by the company and expire after a limited time (even just a few hours!), drawing business into the store and truly capitalizing off of the fast pace of internet connected life.

A Blossom smartphone application should also allow customers to download music while they're in the store. For example, customers could be provided with an easy way to download whatever song is currently playing for free (provided they're physically inside of the store). This would present an opportunity for Blossom to not only get people into stores, but also to promote new music. One could imagine people flocking to a store because they've heard that Blossom will be giving away a popular new song for free, but they would have to wait until that song actually started playing before they could download it. They would, of course, be buying drinks in the process. A Blossom app could also allow people to connect with in-store electronics easily. Imagine if you could sit at a table where the entire surface is a touchscreen, then it recognizes you (because of your cell phone) and pulls up your own personalized interface. Also, the Blossom app could allow people to connect with each other in new ways. By utilizing an Augmented Reality style Heads Up Display, people could look around the store and find information about the people around them, and share some of their own information. That way, a new friendship could be born between two people over a common interest in music or art, without the people just randomly chatting with each other and hoping it would happen. People could even decide what kind of person they wanted to display their information to; for example a teenage girl might have her information be only visible to other teenagers, while a middle-aged man might have his information visible only to other people who work at his company. This protection would allow people to connect with other people like them, and help guard against undesired approaches that are common even without public displays of information. A smartphone application will help Blossom to reach beyond the walls of its stores, beyond the location of its coffee machines, and allow people to interact and engage through Blossom's network wherever they have a cellular phone.

The store also should extend beyond serving just coffee, being open into the night as a casual lounge, serving coffee based alcoholic beverages, and fine wines and beer. The stores should also have an air of exclusivity, something that Starbucks lost through its massive expansion. A good network of stores would place flagship stores in highly visible places in major cities. Each city should have only one flagship store to maintain an air of exclusivity. Flagship stores should be equipped with the newest and best the company has to offer, and be more tuned towards an immersive consumer experience. Each flagship store should include unique aspects of its home city, so that people from that city could feel proud of their store and understand that it is unique to them, yet still part of the international Blossom experience. There could also be satellite stores, much more similar to current Starbucks or even tiny New York coffee bars. Satellite stores would still carry the company name and logo, but be designed for speed so that customers can come, get their coffee, and get on their way. Satellite stores would be more low key, so customers who did not desire to have the high-end visible experience provided at the flagship store could still enjoy high quality coffee and feel connected with the Blossom experience. With such a network of stores, the company could maintain a high brand image while providing different segments of customers with a unique coffee experience that satisfies an as-yet unrealized need.

A possible satellite store design would have the customers brewing their own coffee. Customers come through the door and immediately walk past bins of coffee (see figure 4.5.1). They would grab the size cup they wanted to have, and each cup would have a line on the bottom indicating the required amount of coffee. Customers would fill the cup with that much coffee from bins that they walk past immediately after coming in the store. By having to get their own beans, they would be instantly connected with the smell and sight of the whole beans, and start establishing a connection with their coffee before they even start making it. There would be no formal counter to approach, baristas would roam the shop equipped with mobile cashiers equipment. To make the payment process as simple as possible, all prices would be a multiple of 25 cents so that the baristas would need to carry only quarters to provide change to the customers. The baristas would carry only \$1, \$5, and \$20 bills to make change more simply. Sales tax would be assessed behind the scenes so that customers could comfortable choose what they wanted and not have to deal with charges added after they read the price. After choosing their beans, customers would then be directed to the Blossom machines, where baristas would assist them with grinding the beans and brewing them. This style of store would create a special connection between each customer and the cup they made, and the self-directed brew process could potentially allow a reduction in staff. Every employee at a Blossom store would be respected and provided with very competitive benefits, like employees at Apple or Starbucks stores.

The mission of a great coffee company should actually be more than simply brewing good coffee. The mission of Blossom should be to create technologies that enhance the lives of people everywhere, whether or not they are customers.

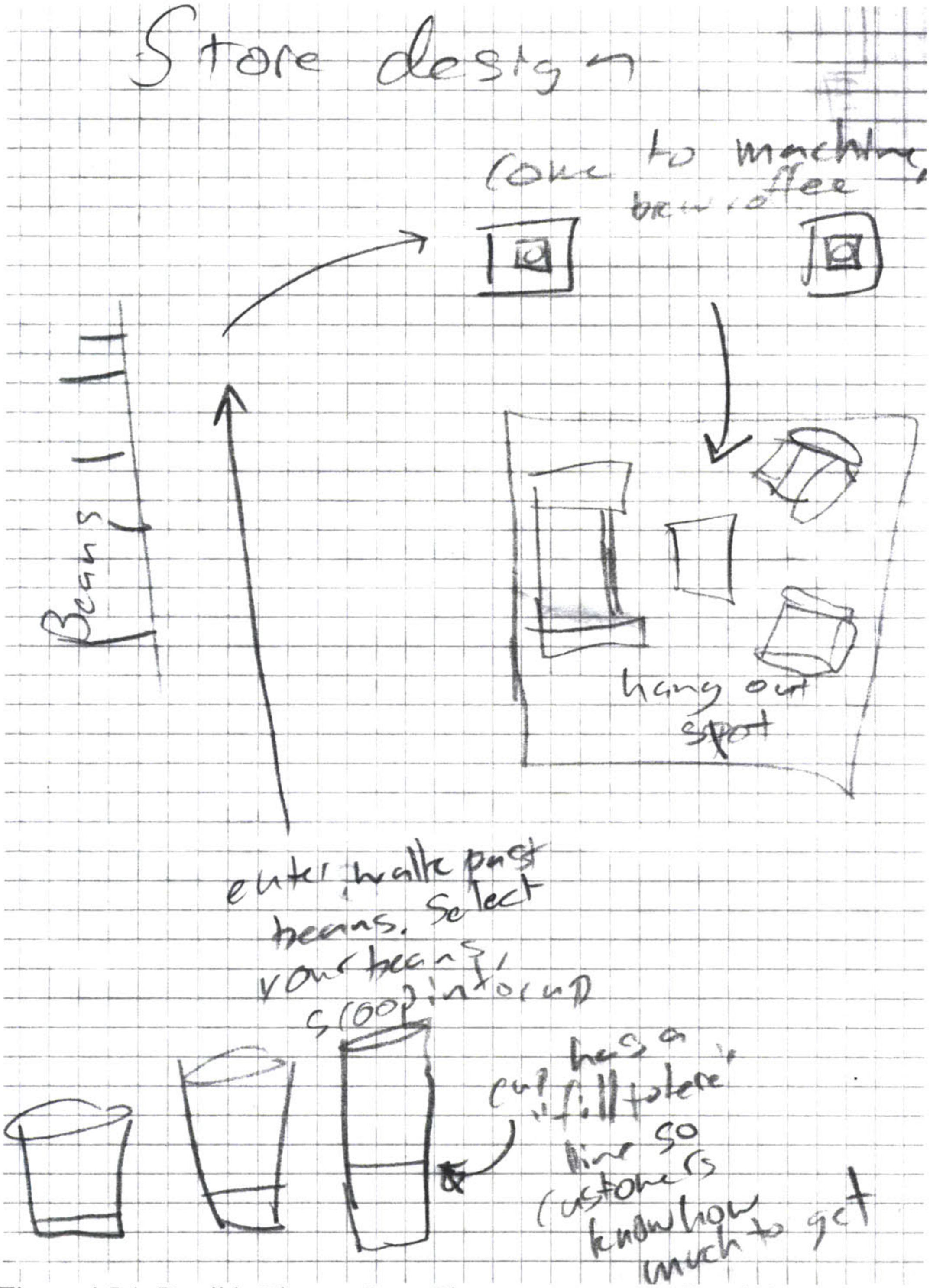
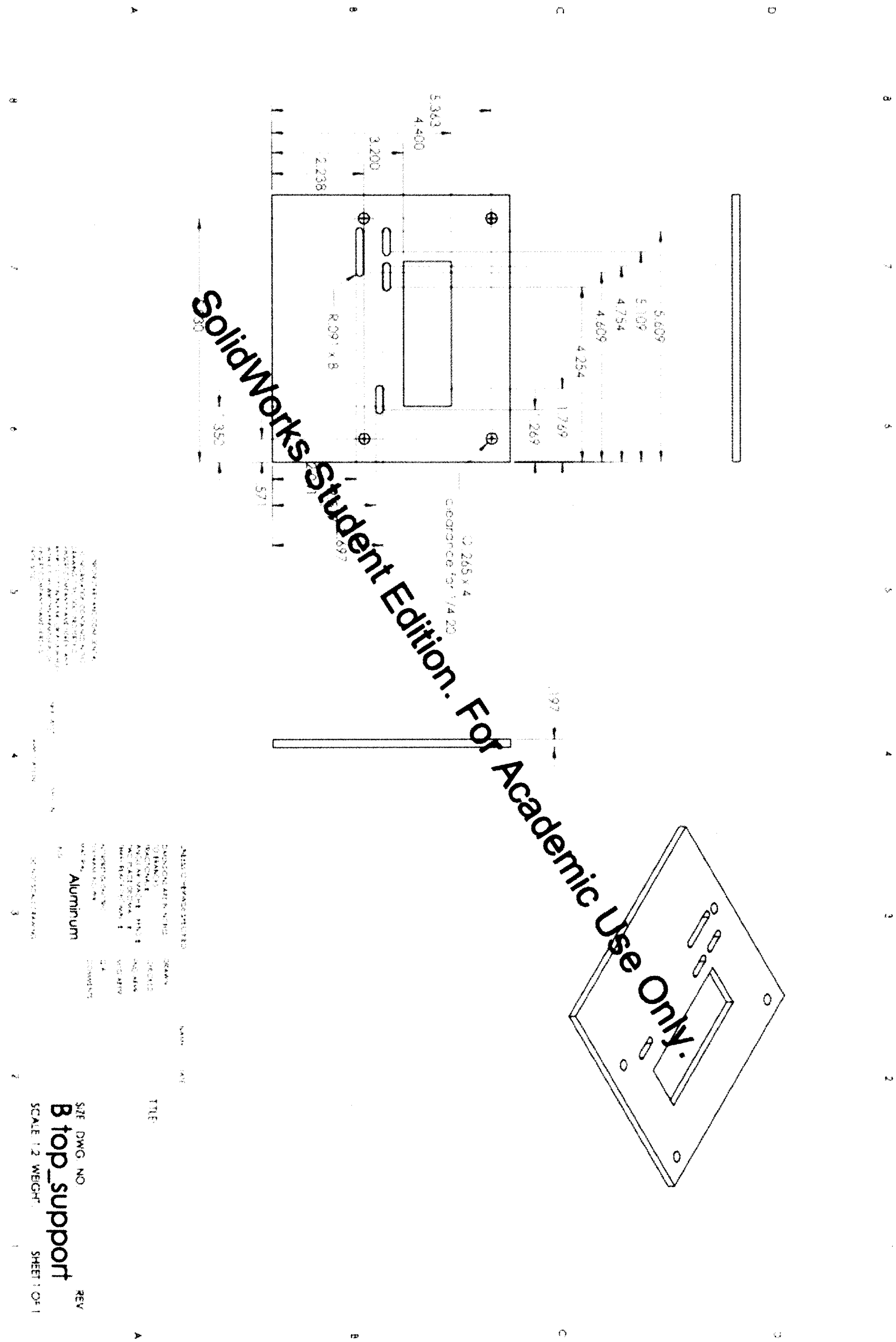
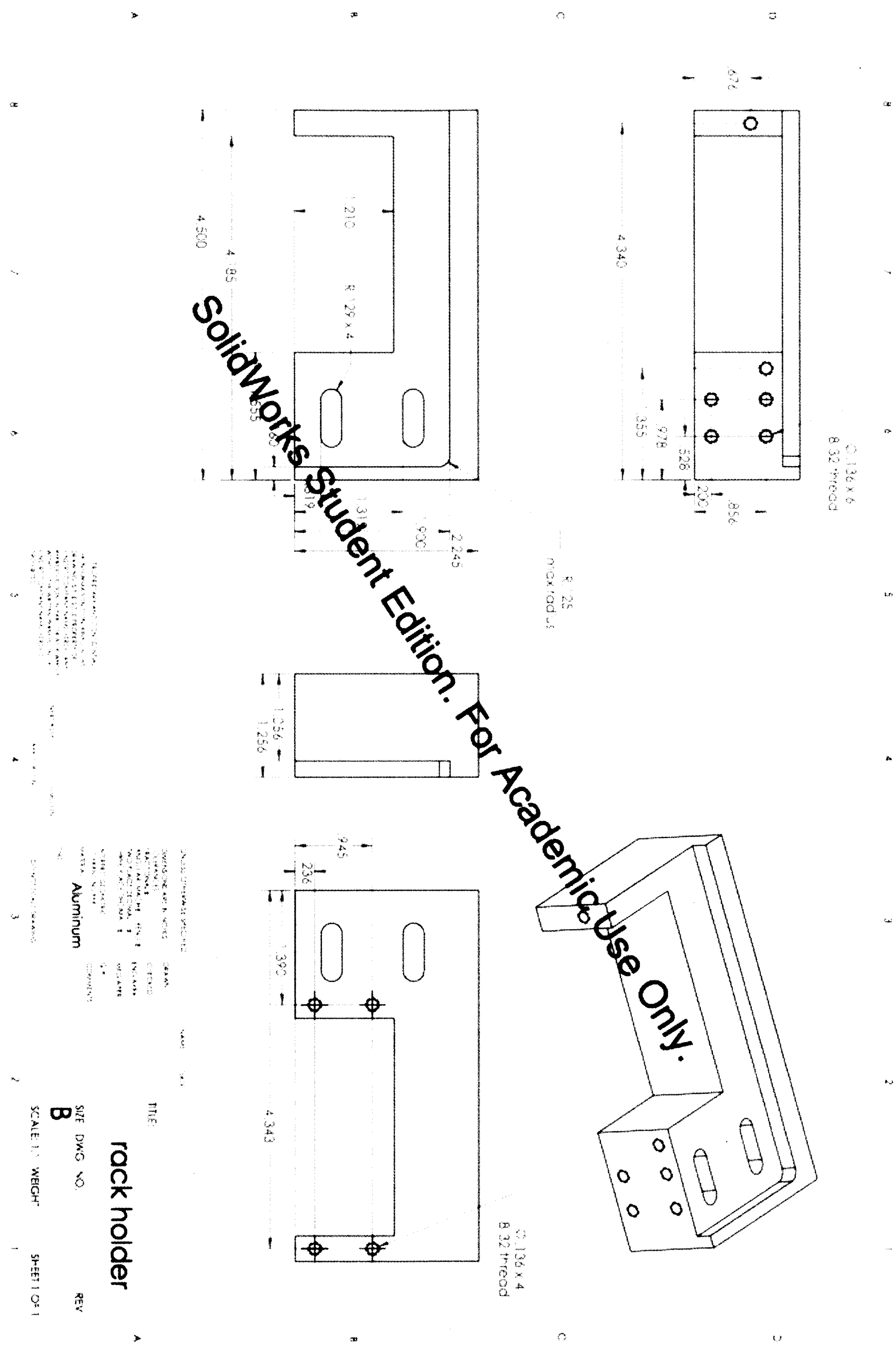


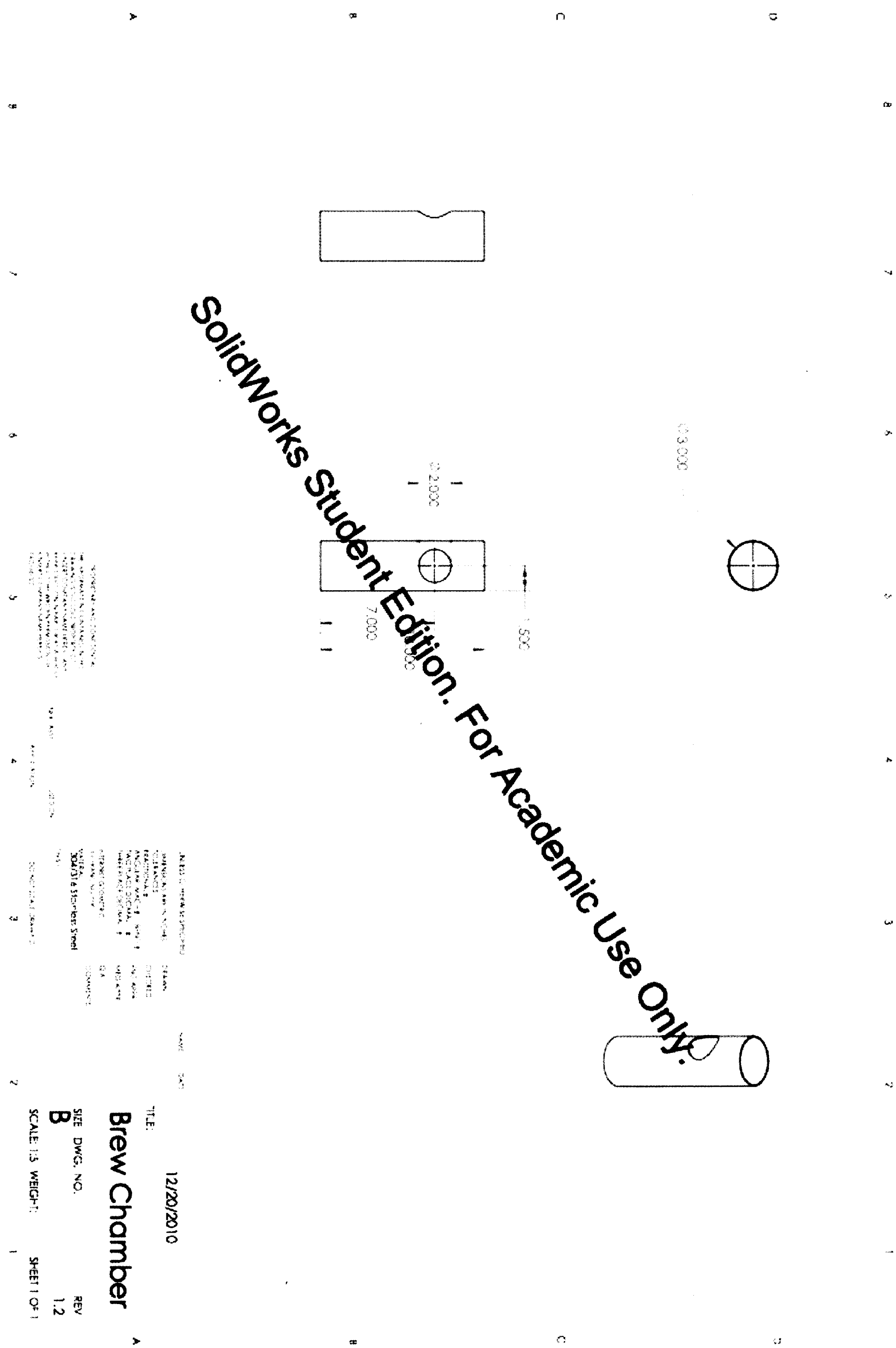
Figure 4.5.1: Possible Blossom Store Diagram; customers select their own beans directly from bulk containers, and put the beans in cups that have pre-marked lines so they know how much to get for each size of cup. After selecting their beans, customers brew their coffee on machines in the store, with the help of a barista. Finally, there is a place for socializing and enjoying coffee. This store design would create a unique experience based around the Blossom machine, and is designed to create a special connection between the customers and the coffee they brew. By having customers brew their own coffee, there is also the potential to increase the rate at which customers are served without increasing the number of baristas present in the store.

Appendix B: Engineering Drawings of Machined Parts

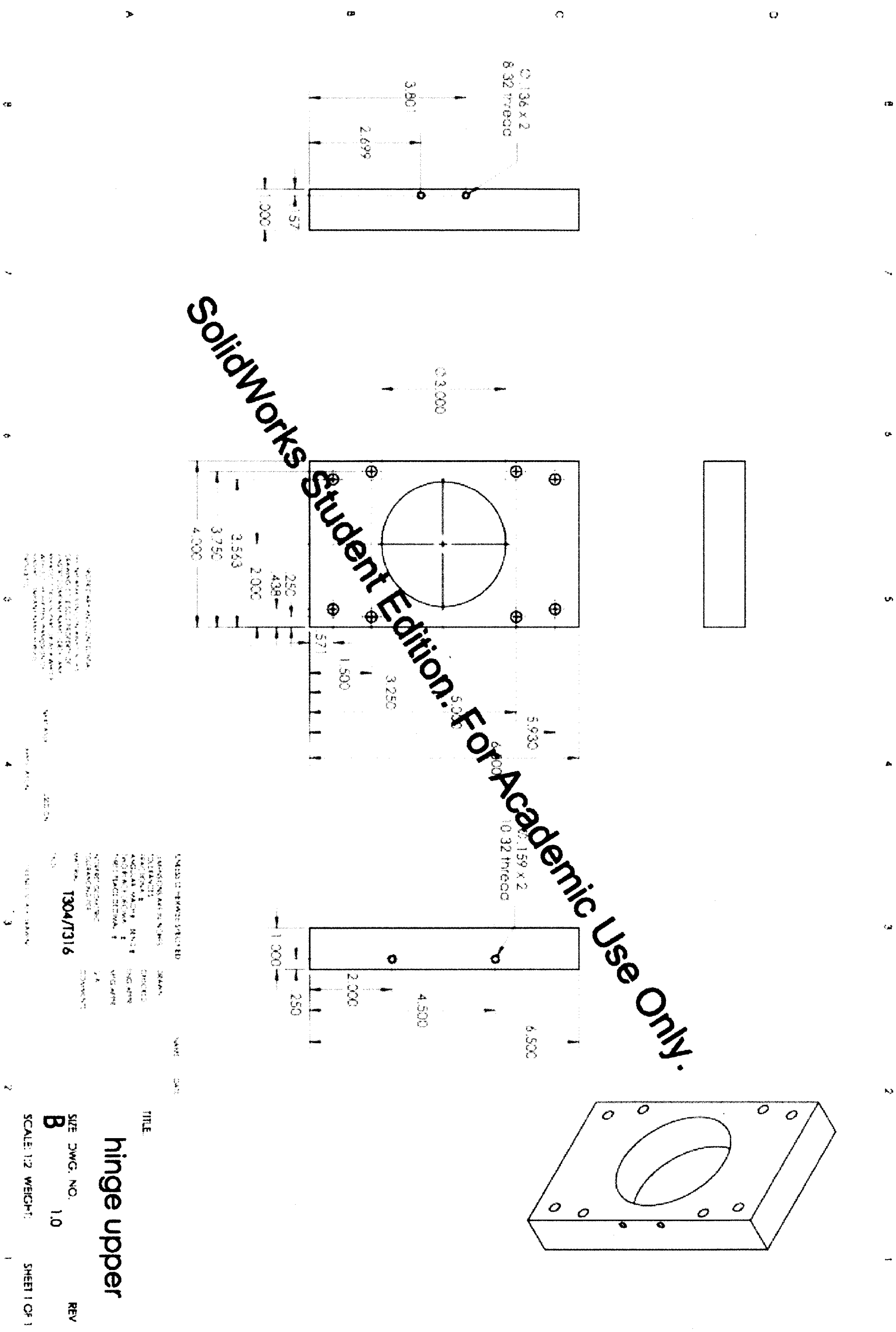


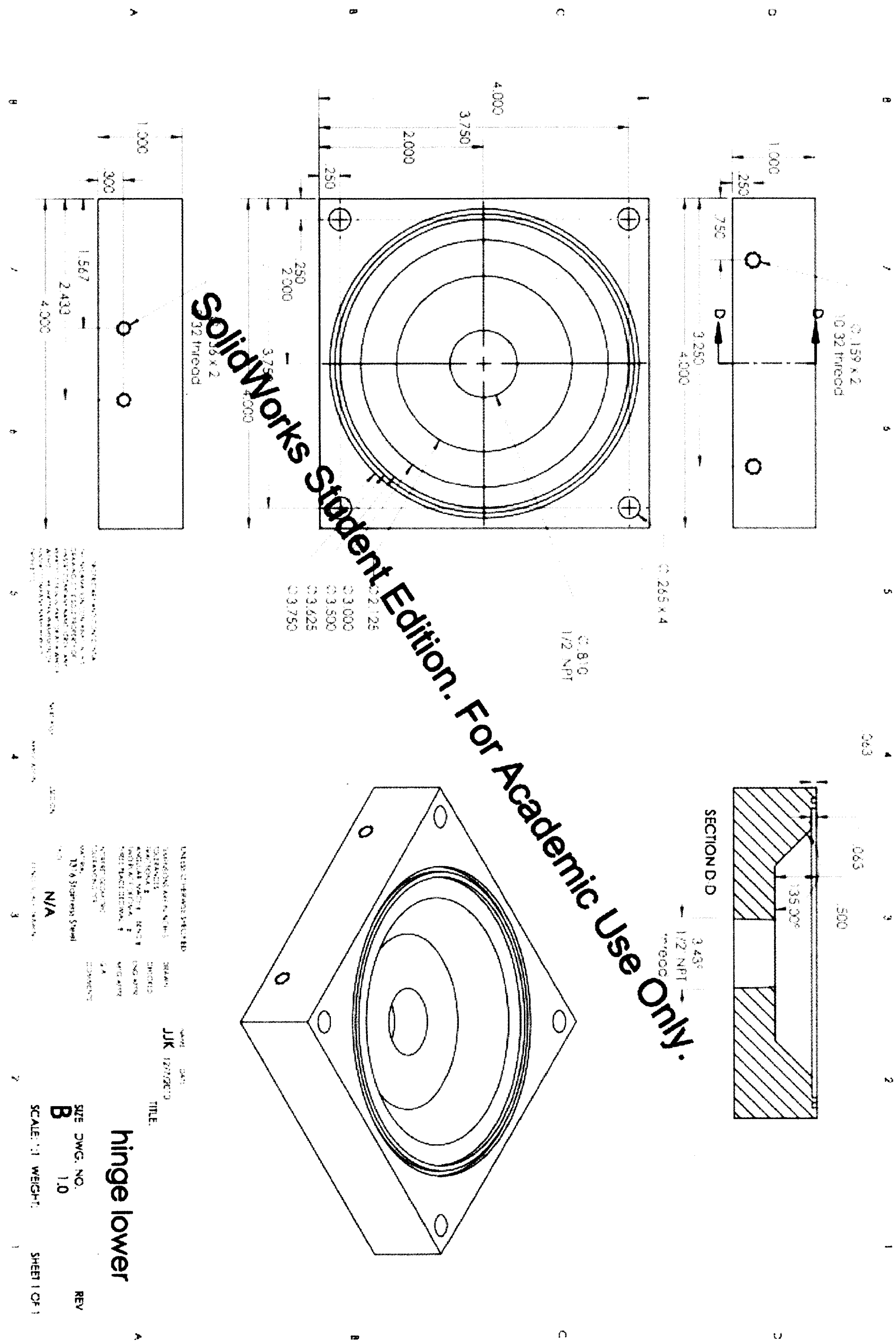


SolidWorks Student Edition. For Academic Use Only.
4.500
1.210
4.340
.626
.355
.978
.526
.200
.856
2.245
.900
1.056
1.256
.945
.236
1.390
4.343
Ø.136 x 6
8-32 thread
Ø.136 x 4
8-32 thread
Aluminum
TITLE:
rack holder
SIZE DWG. NO. REV
B
SHEET 1 OF 1



SolidWorks Student Edition. For Academic Use Only.
7.000
.500
TITLE:
Brew Chamber
12/20/2010
SIZE DWG. NO.
B
REV
1.2
SCALE: 1:5 WEIGHT:
SHEET 1 OF 1
304/316 Stainless Steel


SolidWorks Student Edition. For Academic Use Only.
hinge upper
TITLE
SIZE
B
DWG. NO.
1.0
REV
SCALE: 1:2 WEIGHT:
SHEET 1 OF 1
T304/T316
Ø3.000
8-32 thread
10-32 thread
3.801
2.699
1.000
4.000
3.750
3.563
2.000
.438
.250
1.500
3.250
5.930
6.000
4.500
6.500
2.000


SolidWorks Student Edition. For Academic Use Only.
SECTION D-D
hinge lower
TITLE:
JJK
N/A
SIZE
B
DWG. NO.
1.0
REV
SCALE: 1:1 WEIGHT:
SHEET 1 OF 1
4.000
3.750
2.000
1.000
.250
.300
1.567
2.430
3.250
Ø .265 x 4
Ø .810
1/2 NPT
Ø 2.125
Ø 3.000
Ø 3.500
Ø 3.625
Ø 3.750
.063
.500
135.00°

Appendix C: Arduino Code

This is the code that was used to operate the Arduino and the control electronics. It was developed in the open-source community and is included here purely for illustrative purposes, no claim is being made for ownership.

```
// BBCC Main
// Tim Hirzel
// February 2008
//
// This code has been modified by Jeremy Kuempel, January 2011
//
// Main file for the Bare Bones Coffee Controller PID
// setup for Arduino.
// This project is set up such that each tab acts as a
// "module" or "library" that incporporates some more
// functionality. Each tab correlates
// to a particular device (Nunchuck), protocol (ie. SPI),
// or Algorithm (ie. PID).

// The general rule for any of these tabs/sections is that
// if they include a setup* or update* function, those should be added
// into the main setup and main loop functions. Also, in main loop, and in
// extra code, delays should probably be avoided.
// Instead, use millis() and check for a certain interval to have passed.
//
// All code released under
// Creative Commons Attribution-Noncommercial-Share Alike 3.0


// These are addresses into EEPROM memory. The values to be stores are floats which
// need 4 bytes each. Thus 0,4,8,12,...
#define PGAIN_ADR 0
#define IGAIN_ADR 4
#define DGAIN_ADR 8
#define BREW_TIME 180000

#define ESPRESSO_TEMP_ADDRESS 12
//#define STEAM_TEMP_ADDRESS 12 // steam temp currently not used with bare bones setup

#define PID_UPDATE_INTERVAL 200 // milliseconds


float targetTemp; //current temperature goal
float heatPower; // 0 - 1000 milliseconds on per second

unsigned long lastPIDTime; // most recent PID update time in ms

void setup()
```

```
{
 setupPID(PGAIN_ADR, IGAIN_ADR, DGAIN_ADR ); // Send addresses to the PID module
 targetTemp = readFloat(ESPRESSO_TEMP_ADDRESS); // from EEPROM. load the saved value
 lastPIDTime = millis();
 // module setup calls
 setupHeater();
 setupSerialInterface();
 setupTempSensor();
}

void setTargetTemp(float t) {
 targetTemp = t;
 writeFloat(t, ESPRESSO_TEMP_ADDRESS);
}

float getTargetTemp() {
 return targetTemp;
}


void loop()
{
 // this call interprets characters from the serial port
 // its a very basic control to allow adjustment of gain values, and set temp
 updateSerialInterface();
 updateTempSensor();

 // every second, udpate the current heat control, and print out current status

 // This checks for rollover with millis()

 if (millis() < lastPIDTime) {
  lastPIDTime = 0;
 }

 if ((millis() - lastPIDTime) > PID_UPDATE_INTERVAL) {
  lastPIDTime += PID_UPDATE_INTERVAL;
  heatPower = updatePID(targetTemp, getFreshTemp());
  setHeatPowerPercentage(heatPower);

 }

// if(millis() > BREW_TIME){
//  runServo();
// }

 updateHeater();
}
```

```
// END BBCC Main
```

```
// Brew Time Control
//This will actuate the servo to run the plunger.
// Jeremy Kuempel, January 2011

#include <Servo.h>

Servo myservo; // create servo object to control a servo
        // a maximum of eight servo objects can be created

int pos = 0;   // variable to store the servo position


void runServo(){
 myservo.attach(9); // attaches the servo on pin 9 to the servo object
 myservo.write(180);         // using a HiTec HS-805BB servo that has been modded for continuous
rotation
delay(5000);
  myservo.write(0);
delay(4000);

 myservo.detach();

 while(true){
 }

}
```

```
// Simple extension to the EEPROM library
// Tim Hirzel
// All code released under
// Creative Commons Attribution-Noncommercial-Share Alike 3.0

#include <avr/EEPROM.h>

float readFloat(int address) {
 float out;
 eeprom_read_block((void *) &out, (unsigned char *) address ,4 );
 return out;
}

void writeFloat(float value, int address) {
 eeprom_write_block((void *) &value, (unsigned char *) address ,4);
}
```

```
// END EEPROM Float
```

```
// HeaterControl
// Tim Hirzel
// Dec 2007
//
// This file is for controlling a heater via a solid state zero crossing relay

// since these are zero-crossing relays, it makes sense to just match my local
// AC frequency, 60hz
//
// All code released under
// Creative Commons Attribution-Noncommercial-Share Alike 3.0

#define HEAT_RELAY_PIN 6

float heatcycles; // the number of millis out of 1000 for the current heat amount (percent * 10)

boolean heaterState = 0;

unsigned long heatCurrentTime, heatLastTime;

void setupHeater() {
 pinMode(HEAT_RELAY_PIN , OUTPUT);
}


void updateHeater() {
 boolean h;
 heatCurrentTime = millis();
 if(heatCurrentTime - heatLastTime >= 1000 or heatLastTime > heatCurrentTime) { //second
statement prevents overflow errors
   // begin cycle
   _turnHeatElementOnOff(1); //
   heatLastTime = heatCurrentTime;
 }
 if (heatCurrentTime - heatLastTime >= heatcycles) {
   _turnHeatElementOnOff(0);
 }
}

void setHeatPowerPercentage(float power) {
 if (power <= 0.0) {
   power = 0.0;
 }
 if (power >= 1000.0) {
   power = 1000.0;
```

```
 }
 heatcycles = power;
}

float getHeatCycles() {
 return heatcycles;
}

void _turnHeatElementOnOff(boolean on) {
 digitalWrite(HEAT_RELAY_PIN, on);    //turn pin high
 heaterState = on;
}

// End Heater Control
```

```
// PID control code
// Tim Hirzel
// December 2007

// This is a module that implements a PID control loop
// initialize it with 3 values: p,i,d
// and then tune the feedback loop with the setP etc funcs
//
// this was written based on a great PID by Tim Wescott:
// http://www.embedded.com/2000/0010/0010feat3.htm
//
//
// All code released under
// Creative Commons Attribution-Noncommercial-Share Alike 3.0



#define WINDUP_GUARD_GAIN 100.0

float iState = 0;
float lastTemp = 0;

float pgain;
float igain;
float dgain;

float pTerm, iTerm, dTerm;

int pgainAddress, igainAddress, dgainAddress;

void setupPID(unsigned int padd, int iadd, int dadd) {
 // with this setup, you pass the addresses for the PID algorithm to use to
```

```
 // for storing the gain settings.  This way wastes 6 bytes to store the addresses,
 // but its nice because you can keep all the EEPROM address allocaton in once place.

 pgainAddress = padd;
 igainAddress = iadd;
 dgainAddress = dadd;

 pgain = readFloat(pgainAddress);
 igain = readFloat(igainAddress);
 dgain = readFloat(dgainAddress);
}

float getP() {
 // get the P gain
 return pgain;
}
float getI() {
 // get the I gain
 return igain;
}
float getD() {
 // get the D gain
 return dgain;
}


void setP(float p) {
 // set the P gain and store it to eeprom
 pgain = p;
 writeFloat(p, pgainAddress);
}

void setI(float i) {
 // set the I gain and store it to eeprom
 igain = i;
 writeFloat(i, igainAddress);
}

void setD(float d) {
 // set the D gain and store it to eeprom
 dgain = d;
 writeFloat(d, dgainAddress);
}

float updatePID(float targetTemp, float curTemp)
{
 // these local variables can be factored out if memory is an issue,
 // but they make it more readable
 double result;
```

```
  float error;
  float windupGaurd;

  // determine how badly we are doing
  error = targetTemp - curTemp;

  // the pTerm is the view from now, the pgain judges
  // how much we care about error we are this instant.
  pTerm = pgain * error;

  // iState keeps changing over time; it's
  // overall "performance" over time, or accumulated error
  iState += error;

  // to prevent the iTerm getting huge despite lots of
  //  error, we use a "windup guard"
  // (this happens when the machine is first turned on and
  // it cant help be cold despite its best efforts)

  // not necessary, but this makes windup guard values
  // relative to the current iGain
  windupGaurd = WINDUP_GUARD_GAIN / igain;

  if (iState > windupGaurd)
    iState = windupGaurd;
  else if (iState < -windupGaurd)
    iState = -windupGaurd;
  iTerm = igain * iState;

  // the dTerm, the difference between the temperature now
  //  and our last reading, indicated the "speed,"
  // how quickly the temp is changing. (aka. Differential)
  dTerm = (dgain* (curTemp - lastTemp));

  // now that we've use lastTemp, put the current temp in
  // our pocket until for the next round
  lastTemp = curTemp;

  // the magic feedback bit
  return  pTerm + iTerm - dTerm;
}

void printPIDDebugString() {
  // A  helper function to keep track of the PID algorithm
  Serial.print("PID formula (P + I - D): ");

  printFloat(pTerm, 2);
  Serial.print(" + ");
  printFloat(iTerm, 2);
```

```
 Serial.print(" - ");
 printFloat(dTerm, 2);
 Serial.print(" POWER: ");
 printFloat(getHeatCycles(), 0);
 Serial.print(" ");

}

// END PID
```

```
// With the AD 595, this process is just a matter of doing some math on an
// analog input
//
// Thanks to Karl Gruenewald for the conversion formula
// All code released under
// Creative Commons Attribution-Noncommercial-Share Alike 3.0

// This current version is based on sensing temperature with
// an AD595 and thermocouple through an A/D pin.  Any other
// sensor could be used by replacing this one function.
// feel free to use degrees C as well, it will just give a different
// PID tuning than those from F.
//


#define TEMP_SENSOR_PIN 1

float tcSum = 0.0;
float latestReading = 0.0;
int readCount = 0;
float multiplier;
void setupTempSensor() {
 multiplier = 1.0/(1023.0) * 500.0 * 9.0 / 5.0;
}

void updateTempSensor() {
   tcSum += analogRead(TEMP_SENSOR_PIN); //output from AD595 to analog pin 1
   readCount +=1;
}

float getFreshTemp() {
     latestReading = tcSum* multiplier/readCount+32.0;
     readCount = 0;
     tcSum = 0.0;
  return latestReading;

}
```

```
float getLastTemp() {
  return latestReading;

}

// END Temperature Sensor
```

```
//serialInterface
// Tim Hirzel February 2008
// This is a very basic serial interface for controlling the PID loop.
// thanks to the Serial exampe code

// All code released under
// Creative Commons Attribution-Noncommercial-Share Alike 3.0

#define AUTO_PRINT_INTERVAL 200  // milliseconds
#define MAX_DELTA  100
#define MIN_DELTA  0.01
#define PRINT_PLACES_AFTER_DECIMAL 2  // set to match MIN_DELTA


int incomingByte = 0;
float delta = 1.0;
boolean autoupdate;
boolean printmode = 0;

unsigned long lastUpdateTime = 0;
void setupSerialInterface()  {
  Serial.begin(9600);
  Serial.println("\nWelcome to the BBCC, the Bare Bones Coffee Controller for Arduino");
  Serial.println("Send back one or more characters to setup the controller.");
  Serial.println("If this is your initial run, please enter 'R' to Reset the EEPROM.");
  Serial.println("Enter '?' for help.  Here's to a great cup!");
}

void printHelp() {
  Serial.println("Send these characters for control:");
  Serial.println("<space> : print status now");
  Serial.println("u : toggle periodic status update");
  Serial.println("g : toggle update style between human and graphing mode");
  Serial.println("R : reset/initialize PID gain values");
  Serial.println("b : print PID debug values");
  Serial.println("? : print help");
  Serial.println("+/- : adjust delta by a factor of ten");
  Serial.println("P/p : up/down adjust p gain by delta");
  Serial.println("I/i : up/down adjust i gain by delta");
  Serial.println("D/d : up/down adjust d gain by delta");
  Serial.println("T/t : up/down adjust set temp by delta");
```

```
}

void updateSerialInterface() {
 while(Serial.available()){

  incomingByte = Serial.read();
  if (incomingByte == 'R') {
   setP(30.0); // make sure to keep the decimal point on these values
   setI(0.0);  // make sure to keep the decimal point on these values
   setD(0.0);  // make sure to keep the decimal point on these values
   setTargetTemp(200.0); // here too
  }
  if (incomingByte == 'P') {
   setP(getP() + delta);
  }
  if (incomingByte == 'p') {
   setP(getP() - delta);
  }
  if (incomingByte == 'I') {
   setI(getI() + delta);
  }
  if (incomingByte == 'i') {
   setI(getI() - delta);
  }
  if (incomingByte == 'D') {
   setD(getD() + delta);
  }
  if (incomingByte == 'd' ){
   setD(getD() - delta);
  }
  if (incomingByte == 'T') {
   setTargetTemp(getTargetTemp() + delta);
  }
  if (incomingByte == 't') {
   setTargetTemp(getTargetTemp() - delta);
  }
  if (incomingByte == '+') {
   delta *= 10.0;
   if (delta > MAX_DELTA)
    delta = MAX_DELTA;
  }
  if (incomingByte == '-') {
   delta /= 10.0;
   if (delta < MIN_DELTA)
    delta = MIN_DELTA;

  }
```

```
  if (incomingByte == 'u') {
    // toggle updating

    autoupdate = not autoupdate;
  }
  if (incomingByte == 'g') {
    // toggle updating

    printmode = not printmode;
  }
  if (incomingByte == ' ') {
    // toggle updating

    printStatus();
  }
  if (incomingByte == '?') {
    printHelp();
  }
  if (incomingByte == 'b') {
    printPIDDebugString();
    Serial.println();
  }
 }

 if (millis() < lastUpdateTime) {
   lastUpdateTime = 0;
 }
 if ((millis() - lastUpdateTime) > AUTO_PRINT_INTERVAL) {
   // this is triggers every slightly more than a second from the delay between these two millis() calls
   lastUpdateTime += AUTO_PRINT_INTERVAL;
   if (autoupdate) {
     if (printmode) {
       printStatusForGraph();
     }
     else {
       printStatus();
     }
   }
 }
}

void printStatus() {
  // A means for getting feedback on the current system status and controllable parameters
  Serial.print(" SET TEMP:");
  printFloat(getTargetTemp(),PRINT_PLACES_AFTER_DECIMAL);
  Serial.print(", CUR TEMP:");
  printFloat(getLastTemp(),PRINT_PLACES_AFTER_DECIMAL);

  Serial.print(", GAINS p:");
```

```
 printFloat(getP(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(" i:");
 printFloat(getI(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(" d:");
 printFloat(getD(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(", Delta: ");
 printFloat(delta,PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(", Power: ");
 printFloat((float)getHeatCycles(), 0);

 Serial.print(" \n");
}

void printStatusForGraph() {
 printFloat(getTargetTemp(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(", ");
 printFloat(getLastTemp(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(", ");
 printFloat(getP(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(", ");
 printFloat(getI(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(", ");
 printFloat(getD(),PRINT_PLACES_AFTER_DECIMAL);
 Serial.print(", ");
 printFloat((float)getHeatCycles(), 0);
 Serial.println();
}

// printFloat prints out the float 'value' rounded to 'places' places after the decimal point
void printFloat(float value, int places) {
 // this is used to cast digits
 int digit;
 float tens = 0.1;
 int tenscount = 0;
 int i;
 float tempfloat = value;

 // make sure we round properly. this could use pow from <math.h>, but doesn't seem worth the import
 // if this rounding step isn't here, the value  54.321 prints as 54.3209

 // calculate rounding term d:   0.5/pow(10,places)
 float d = 0.5;
 if (value < 0)
   d *= -1.0;
 // divide by ten for each decimal place
 for (i = 0; i < places; i++)
   d/= 10.0;
 // this small addition, combined with truncation will round our values properly
 tempfloat +=  d;
```

```
  // first get value tens to be the large power of ten less than value
  // tenscount isn't necessary but it would be useful if you wanted to know after this how many chars the
number will take

  if (value < 0)
    tempfloat *= -1.0;
  while ((tens * 10.0) <= tempfloat) {
    tens *= 10.0;
    tenscount += 1;
  }


  // write out the negative if needed
  if (value < 0)
    Serial.print('-');

  if (tenscount == 0)
    Serial.print(0, DEC);

  for (i=0; i< tenscount; i++) {
    digit = (int) (tempfloat/tens);
    Serial.print(digit, DEC);
    tempfloat = tempfloat - ((float)digit * tens);
    tens /= 10.0;
  }

  // if no places after decimal, stop now and return
  if (places <= 0)
    return;

  // otherwise, write the point and continue on
  Serial.print('.');

  // now write out each decimal place by shifting digits one by one into the ones place and writing the
truncated value
  for (i = 0; i < places; i++) {
    tempfloat *= 10.0;
    digit = (int) tempfloat;
    Serial.print(digit,DEC);
    // once written, subtract off that digit
    tempfloat = tempfloat - (float) digit;
  }
}

// END Serial Interface
```

The following code was written in the open source community to run in the Processing environment. It codes for the View, a screenshot can be seen in Figure 6.4.3. Again, it is being included purely for illustrative purposes, no claim is being made to ownership.

```
/*
BBCC_Plotter
 by Tim Hirzel, Feb 2008
 v 1.1

 with gratitude, based on:
 Grapher Pro!
 by Tom Igoe


 This Processing application is designed to plot and display values
 coming from the BBCC PID controller for Arduino.  With minimal modification,
 it could be setup to plot and show different incoming data

 Please refer to the top section for alterations of plot ranges, graph size etc.

 All code released under
 Creative Commons Attribution-Noncommercial-Share Alike 3.0
 */

import processing.serial.*;

// *********  unlikely that you want to change these  *********
int BAUDRATE = 9600;
char DELIM = ','; // the delimeter for parsing incoming data
char INIT_MARKER = '!';     //Look for this character before sending init message

String INIT_MESSAGE = "?gu";  //Send these characters to the arduino after init to start up messages
// this gets the help string from teh device, and then turns on plotting mode with constant update strings

//  *********  SETINGS  *********

int yBorder = 60;  // Size of the background area around the plot in screen pixels
int xBorder = 90;

// the plot size in screen pixels
int plotHeight = 450;
int plotWidth = 750;

int ExpectUpdateSpeed = 200; // milliseconds.  This just allows the axis labels in the X direction
accurate

// These are all in real number space
// all X values measured in ExpectedUpdateSpeed Intervals
```

```
// all y measured in degrees
int gridSpaceX = 50;
int gridSpaceY = 50;
int startX = 0;
int endX = 600;
int startY = 0;
int endY = 250;

// leave these to be calculated
float pixPerRealY = float(plotHeight)/(endY - float(startY));
float pixPerRealX = float(plotWidth)/(endX - float(startX));

// These are calculated here, but could be changed if you wanted
int windowWidth = plotWidth + 2*xBorder;
int windowHeight = plotHeight +  2*yBorder;

// ******* Legend  ********
// Define the location and size of the Legend area
int legendWidth = 125;
int legendHeight = 130;
int legendX = windowWidth - 140;
int legendY =  15;

// ******* Help Window  ********
// Define the size of the help area.  It always sits in the middle
int helpWidth = 600;
int helpHeight = 400;


String title = "Bare Bones Coffee Controller Tuning";   // Plot Title
String names = "Goal Curr P I D Pow";                    // The names of the values that get sent over serial
String yLabel = "T\ne\nm\np\ne\nr\na\nt\nu\nr\ne\n\nF";  // this is kind of a hack to make the vertical
label
String xLabel = "Seconds";                   // X axis label
String fontType = "Courier";                 // Y axis label
boolean[] plotThisName = {
  true,true, false, false, false, false};            // For each of the values, you can choose if you want it
plotted or not here

// ****************  end of Settings area  ****************

String helpBase = "-Plotter Help-\n(all characters are case sensitive)\nh : toggle this help screen\nl :
toggle the Legend\nS : save screen\nE : export shown data as text\n\n-Device Help-  \n";
String helpString = "";
Serial myPort;            // The serial port

boolean displayLegend = true;
boolean displayHelp = true;
```

```
int sensorCount = 6;                    // number of values to expect
float[][] sensorValues = new float[endX-startX][sensorCount];  // array to hold the incoming values
int currentValue = -1;
int hPosition = startX;                    // horizontal position on the plot
int displayChannel = 0;                  // which of the five readings is being displayed
int threshold = 50;                   // threshold for whether or not to write
// data to a file
boolean updatePlot = false;
//int [] lastSet  = new int[sensorCount];

int[][] colors = new int[sensorCount][3];

PFont titleFont;
PFont labelFont;

void setupColors() {
  // Thanks to colorbrewer for this pallete
  colors[0][0] = 102;
  colors[0][1] =194;
  colors[0][2] = 165;
  colors[1][0] = 252;
  colors[1][1] = 141;
  colors[1][2]= 98;
  colors[2][0] = 141;
  colors[2][1] = 160;
  colors[2][2]= 203;
  colors[3][0] = 231;
  colors[3][1] = 138;
  colors[3][2]= 195;
  colors[4][0] = 166;
  colors[4][1] = 216;
  colors[4][2]= 84;
  colors[5][0] = 255;
  colors[5][1] = 217;
  colors[5][2]= 47;
}


void setup () {
  size(windowWidth, windowHeight);        // window size
  setupColors();
  smooth();
  //  println(PFont.list());
  titleFont = createFont(fontType, 18);
  labelFont = createFont(fontType, 14 );

  clearPlot();
  // List all the available serial ports
  println(Serial.list());
```

```
  // On my mac, the arduino is the first on this list.
  // Open whatever port is the one you're using.
  myPort = new Serial(this, Serial.list()[0], BAUDRATE);
  // clear the serial buffer:
  myPort.clear();
}

void draw () {
  // if the value for the given channel is valid, plot it:
  if (updatePlot) {
    // draw the plot:
    plot();
    updatePlot = false;
  }
}

void clearPlot() {
  background(5);
  strokeWeight(1.5);
  stroke(10);
  fill(40);
  // draw boundary
  rect(xBorder,yBorder,plotWidth, plotHeight);

  textAlign(CENTER);
  fill(70);
  textFont(titleFont);
  text(title, windowWidth/2, yBorder/2);

  textFont(labelFont);
  stroke(10);
  //draw grid
  fill(70);
  textAlign(RIGHT);
  for (int i = startY; i <= endY; i+= gridSpaceY) {
    line(xBorder - 3, realToScreenY(i), xBorder + plotWidth - 1,  realToScreenY(i));
    text(str(i), xBorder - 10, realToScreenY(i));
  }

  textAlign(LEFT);
  for (int i = startX; i <= endX ; i+= gridSpaceX) {
    line(realToScreenX(i), yBorder+1, realToScreenX(i), yBorder + plotHeight + 3);
    text(str((i)/ (1000 / ExpectUpdateSpeed)), realToScreenX(i), yBorder + plotHeight + 20);
  }

  // Draw Axis Labels
  fill(70);
  text(yLabel, xBorder - 70,  yBorder + 100 );
```

```
  textAlign(CENTER);
  text(xLabel,  windowWidth/2, yBorder + plotHeight + 50);


}

float realToScreenX(float x) {
  float shift = x - startX;
  return (xBorder + shift * pixPerRealX);
}

float realToScreenY(float y) {
  float shift = y - startY;
  return yBorder + plotHeight - 1 - (shift) * pixPerRealY;

}

void plot () {
  clearPlot();
  // draw the line:
  for (int i = 0; i < sensorCount; i++) {
    // assign color to each plot
    stroke(colors[i][0], colors[i][1],colors[i][2]);

    for (int x = 1; x < currentValue; x++) {
      if(plotThisName[i]) {

        line(realToScreenX(x-1),
        realToScreenY(sensorValues[x-1][i]) ,
        realToScreenX(x),
        realToScreenY(sensorValues[x][i])
          );

      }
    }
  }

  if (hPosition >= endX) {
    hPosition = startX;
    // wipe the screen clean:
    clearPlot();
  }
  else {
    hPosition += 1;
  }


  noStroke();
```

```
  // DRAW LEGEND
  if (displayLegend) {
    fill(128,128,128,80);
    rect(legendX, legendY, legendWidth, legendHeight);

    // print the name of the channel being graphed:
    String line;
    for (int i = 0; i < sensorCount; i++) {
      fill(colors[i][0], colors[i][1],colors[i][2]);
      textAlign(LEFT);
      text(split(names,' ')[i] , legendX+5, legendY + (i+1) * 20);
      textAlign(RIGHT);
      text(nf(sensorValues[currentValue][i], 0,3), legendX+legendWidth - 5, legendY + (i+1) * 20);
    }
  }

  if (displayHelp) {
    textAlign(LEFT);
    fill(128,128,128,80);
    noStroke();
    rect(windowWidth/2 - helpWidth/2, windowHeight/2 - helpHeight / 2, helpWidth, helpHeight);
    fill(255,255,255);
    helpWidth -= 20;
    helpHeight -=20;
    text(helpString,windowWidth/2 - helpWidth/2, windowHeight/2 - helpHeight / 2, helpWidth,
helpHeight);
    helpWidth += 20;
    helpHeight +=20;

  }
}

void keyPressed() {
  // if the key pressed is "0" through "4"
  if (key == 'l') {
    // set the display channel accordingly
    displayLegend = ! displayLegend;
    updatePlot = true;
  }
  if (key == 'h') {
    // set the display channel accordingly
    displayHelp = ! displayHelp;
    updatePlot = true;
  }
  if (key == 'S') {
    // set the display channel accordingly
    save(str(hour()) + "h" + str(minute()) + "m" + str(second()) +  "s"  + str(month()) + "." + str(day()) +
"." + str(year())+".jpg") ;
  }
```

```
 if (key == 'E') {
   exportText();
 }
 myPort.write(key);

}

void exportText() {
 // string for the new data you'll write to the file:
 String[] outStrings = new String[currentValue+1];
 outStrings[0] = names;
 for (int i =0; i < currentValue; i++) {
   outStrings[i+1] = "";
   for (int j=0; j < sensorCount; j++) {
     outStrings[i+1]   += str(sensorValues[i][j]);
     if (j < sensorCount - 1) {
       outStrings[i+1] += ", ";
     }
   }
 }
 saveStrings(str(hour()) + "h" + str(minute()) + "m" + str(second()) +  "s"  + str(month()) + "." +
str(day()) + "." + str(year())+".txt", outStrings);
}

// make up a timeStamp string for writing data to the file:
String timeStamp() {
 String now = hour()+ ":" +  minute()+ ":" + second()+ " " +
   month() + "/"  + day() + "/" + year();
 return now;
}


void serialEvent(Serial myPort) {
 // read incoming data until you get a newline:
 String serialString = myPort.readStringUntil('\n');
 // if the read data is a real string, parse it:

 if (serialString != null) {
   //println(serialString.charAt(serialString.length()-3));
   // println(serialString.charAt(serialString.length()-2));
   if ((serialString.length() > 2) && (serialString.charAt(serialString.length()-3) == INIT_MARKER))
{
     myPort.write(INIT_MESSAGE);
     helpString  = helpBase;

   }
   else {
     // split it into substrings on the DELIM character:
     String[] numbers = split(serialString, DELIM);
```

```
    // convert each subastring into an int
    if (numbers.length == sensorCount) {
      currentValue += 1;
      if (currentValue >= (endX-startX))
      {
        currentValue = 0;
      }
      for (int i = 0; i < numbers.length; i++) {
        // make sure you're only reading as many numbers as
        // you can fit in the array:
        if (i <= sensorCount) {
          // trim off any whitespace from the substring:
          numbers[i] = trim(numbers[i]);
          sensorValues[currentValue][i] =  float(numbers[i]);
        }
      }
      updatePlot = true;
    }
    else if (currentValue == -1){
      // The help string from the first '?' character gets appended to the plotter help string
      helpString += serialString;
    }
    else {
      // Things we don't handle in particular can get output to the text window
      print(serialString);
    }
  }
 }
}
```